

2003

# On-line optimization of thermomechanical pulping rejects screen using modified sequential simplex method

Hennings, Jeffery Richard

---

<http://knowledgecommons.lakeheadu.ca/handle/2453/3245>

*Downloaded from Lakehead University, Knowledge Commons*

# **On-line Optimization of Thermomechanical Pulping Rejects Screen using Modified Sequential Simplex Method**

**Jeff Hennings**

**A THESIS SUBMITTED IN PARTIAL FULFILMENT OF  
THE REQUIREMENTS OF MscEng DEGREE  
IN  
CONTROL ENGINEERING  
FACULTY OF ENGINEERING  
LAKEHEAD UNIVERSITY  
THUNDER BAY, ONTARIO**

**August 2003**



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file   Votre référence*

*ISBN: 0-612-96978-9*

*Our file   Notre référence*

*ISBN: 0-612-96978-9*

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

# Canada

## **Abstract**

The sequential simplex method (SSM) was blended with the Nelder and Mead method to create two separate simplex methods. The first method is the Equilateral Optimum Following Simplex (EOFS). In this method, the simplex remains equilateral during all steps and changes in size. The other method created in this project is the Non-equilateral Optimum Following Simplex (NOFS). This method is more like the Nelder and Mead method because it has the ability to change its shape when expanding or reducing its size. Both of the methods differ from their predecessor because they are being applied to a system with a moving optimum. Neither the SSM or the Nelder and Mead method have the ability to recognize movement in the location of the optimum. Both methods are excellent at searching for a static optimum, but are incapable of searching for a moving optimum. If either of the SSM or Nelder and Mead methods began to cycle around a single point they would fail to recognize if the optimum moved outside of the current simplex circular area. Both of the methods would continue to focus on the current best point. The best point must be periodically re-evaluated to ensure it isn't out-dated. When the simplex is ready to contract in both the EOFS and NOFS methods, instead of replacing the worst point, the best point is replaced. This does slow the immediate progress of the simplex, but ensures the long-term success in searching out a moving optimum.

Once the simplex methods had been refined on paper, testing each of them in a computer simulation against three types of disturbances was important in determining their potential in an industrial setting. Each simplex method was tested against feed flow, feed fibre distribution and feed consistency disturbances. Computer simulations were run after creating a model of the pressure screen using experimental data obtained from a refined rejects pressure screen at the Thunder Bay, Ontario mill owned by Bowater Inc. Samples were obtained at a variety of operating conditions and analyzed using the Kajaani Fibre Lab to create fibre distributions. These distributions were used to create the model of the screen and the fibre distribution of the feed stream to the screen.

The NOFS simplex method performed well when using two different types of performance equations. One equation was based on achieving maximum fibre separation in the screen, while the other was based on obtaining a target fibre distribution.

The NOFS method proved to be the most effective method because of its ability to accelerate towards the optimum by changing shape. In some simulations noise was added to determine its effect. Noise caused poor results, but further modifications to the simplex methods would probably solve this problem in future studies.

## **Acknowledgements**

I would like to thank Dr. Allan Gilbert for his patience through the project and expertise on the subject. I would also like to thank Geoff Hill of Bowater Inc. for allowing me to study and take samples from the thermomechanical pulping refined rejects pressure screen. I would also like to thank Jamie Puumala, formerly of Bowater Inc., for his assistance in coordinating work and obtaining lab analysis of samples. I would also like to thank the staff of the control room and the lab in the TMP at Bowater for their assistance.

## Table of Contents

Abstract.....	i
Acknowledgements.....	iii
Table of Contents.....	iv
List of Figures.....	v
List of Tables.....	vii
 1. Introduction.....	 1
1.1 TMP Screening.....	1
1.1.1 TMP Manufacture.....	1
1.1.2 Pressure Screening.....	2
1.1.3 Pressure Screen Models.....	10
1.1.4 Plug-Flow Model.....	12
1.1.5 Mixed-Flow Model.....	18
1.1.6 Freeness Model.....	20
1.2 Process Optimization.....	22
1.2.1 Sequential Simplex Method.....	22
1.2.2 Nelder and Mead Method.....	29
1.2.3 Dynamic Simplex Method.....	30
1.2.4 Modifications required for direct control applications.....	31
1.3 Pressure Screen Optimization.....	31
 2. The Modified Simplex Methods.....	 33
2.1 The Non-equilateral Optimum Following Simplex.....	33
2.2 Optimization of a Known Equation.....	42
2.3 Optimization using the Equilateral Optimum Following Simplex.....	63
 3. Optimization of TMP Screening.....	 72
3.1 Reject Screen and Instrumentation.....	72
3.2 Experimental Data.....	73
3.3 The Screening Cost Function.....	82
3.4 Reject Screen Optimization.....	95
3.5 Comparing the Optimum Following Simplex Methods during Disturbances.....	147
 Conclusions.....	 149
 Recommendations.....	 151
 References.....	 152
 Appendix A: simplex1.c	
Appendix B: seqsimp.c	
Appendix C: simp.c	

## List of Figures

1.1	A flow diagram of refining and screening stages in a TMP plant .....	3
1.2	Schematic diagram of the rejects pressure screen showing the feed, shower, accepts and rejects .....	4
1.3	A typical pulp screen .....	5
1.4	The screen rotor and foil blades .....	8
1.5	Cross-sectional view of the screen rotor design .....	8
1.6	Screen cross-section showing the flows through the screen for plug-flow model .....	15
1.7	Plot of fibre passage ratio from Olson et al .....	16
1.8	Reject fibre thickening factor for the plug flow method .....	19
1.9	Reject consistency increases exponentially as reject ratio decreases .....	19
1.10	Schematic diagram of the mixed flow screen model .....	21
1.11	Flow diagram of the sequential simplex method .....	25
1.12	Reflection of a simplex away from worst point .....	26
1.13	Contraction of equilateral simplex towards optimum location .....	28
1.14	Flow diagram of the Nelder and Mead method of optimization .....	32
2.1	Flow diagram of the NOFS method .....	34
2.2	Elongated simplex illustrating the location of the centroid .....	36
2.3	Expansion of the simplex using the NOFS method .....	37
2.4	Contraction of the simplex by replacing the worst point using the NOFS method .....	40
2.5	Contraction of the simplex by replacing the best point using the NOFS method .....	41
2.6	Use of the NOFS method to search out a known optimum .....	43
2.7	Searching for the moving optimum .....	44
2.8	The effect of a larger starting simplex size when searching for the optimum .....	46
2.9	The effect of a larger minimum simplex size when searching for the optimum .....	47
2.10	The effect of a larger maximum simplex size when searching for the optimum .....	51
2.11	Tracking the progress of the simplex in Figure 2.10 .....	52
2.12	The effect of using a larger size ratio for expansion/contraction when searching for the optimum ...	60
2.13	Illustrating the movements of the simplex for the simulation shown in Figure 2.12 for the time period 0 to 25 samples .....	61
2.14	The effect of using a larger size ratio for expansion/contraction and a larger maximum simplex size when searching for the optimum .....	62
2.15	Searching for a static optimum using the EOFS method .....	65
2.16	Searching for a moving optimum using the EOFS method .....	66
2.17	Searching for a moving optimum using the EOFS method and a larger starting simplex .....	67
2.18	Searching for a moving optimum using the EOFS method and a larger minimum simplex size .....	68
2.19	Searching for a moving optimum using the EOFS method and a larger maximum simplex size .....	69
3.1	A diagram of the reject pressure screen and its control structure .....	74
3.2	Instrument loop diagram for the differential pressure control across the screen .....	75
3.3	Instrument loop diagram for the shower flow control on the screen .....	76
3.4	Instrument loop diagram for the reject flow control on the screen .....	77
3.5	Experimental feed fibre size distribution .....	80
3.6	Fibre passage ratio calculated from experimental data .....	81
3.7	Surface plot of the fibre separation cost equation .....	86
3.8	Cost surface for the target fibre distribution equation .....	87
3.9	Cost surface for the accepts freeness equation .....	88
3.10	Cost surface for the rejects consistency .....	89
3.11	Surface for the overall cost equation without boundaries on the valves .....	90
3.12	Contour plot of the overall cost equation without the valve boundaries .....	91
3.13	Cost surface for the valve boundaries .....	92
3.14	Surface for the overall cost equation .....	93
3.15	Contour plot for the overall cost equation .....	94
3.16	Trend plots for the simulation Trial EOFS #1 .....	100
3.17	Trend plots for the simulation Trial EOFS #2 .....	102
3.18	Trend plots for the simulation Trial EOFS #3 .....	104



3.19	Trend plots for the simulation Trial NOFS #1.....	106
3.20	Trend plots for the simulation Trial NOFS #2.....	108
3.21	Trend plots for the simulation Trial NOFS #3.....	110
3.22	Trend plots for the simulation Trial NOFS #4.....	112
3.23	Trend plots for the simulation Trial EOFs #4.....	118
3.24	Trend plots for the simulation Trial EOFs #5.....	120
3.25	Trend plots for the simulation Trial NOFS #5.....	122
3.26	Trend plots for the simulation Trial NOFS #6.....	124
3.27	Trend plots for the simulation Trial NOFS #7.....	126
3.28	Trend plots for the simulation Trial NOFS #8.....	128
3.29	Trend plots for the simulation Trial NOFS #8a.....	130
3.30	Trend plots for the simulation Trial EOFs #6.....	135
3.31	Trend plots for the simulation Trial EOFs #7.....	137
3.32	Trend plots for the simulation Trial NOFS #9.....	139
3.33	Trend plots for the simulation Trial NOFS #10.....	141
3.34	Trend plots for the simulation Trial NOFS #11.....	143
3.35	Trend plots for the simulation Trial NOFS #12.....	145

## List of Tables

3.1	Fibre size categories and ranges of length.....	73
3.2	Fibre size distribution for accepts and rejects using the plug flow method.....	78
3.3	Fibre size distribution for accepts and rejects using the mixed-flow method.....	79
3.4	Fibre size distribution for accepts and rejects from experimental data.....	79
3.5	Combinations of the conditions used to create the simulations for the EOFS method.....	98
3.6	Combinations of the conditions used to create the simulations for the NOFS method .....	98

## **1. Introduction**

Optimization of a thermomechanical pulping (TMP) screen using a modified sequential simplex method is a new idea in the pulp and paper industry. Although sophisticated instrumentation is available and advanced controls are used, this type of optimization has never been attempted in the pulp and paper industry. If optimizing the refined rejects pressure screen is successful, the results of this project could be applied towards optimizing the entire TMP screening system, along with other processes within the pulp and paper industry.

The sequential simplex method was chosen as the optimization technique because it doesn't require a model of the plant in order to search out the optimum. This chapter explains the operation of a TMP screen, analytical models of the screen, and the optimization techniques used in the project. Although the sequential simplex method doesn't require a model of the screen, a model is required when testing the optimization through computer simulation.

### **1.1 TMP Screening**

#### **1.1.1 TMP Manufacture**

A mechanical pulping system, in general, consists of primary and secondary refining stages followed by primary and secondary screening stages. The pulping stages develop the fibre characteristics while the screening stages ensures that only the acceptable pulp goes to the paper machine and that all the components which require further development are directed to the rejects refining stage [1]. After secondary screening the rejected fibres are further refined and screened before going to the paper machine. A major type of mechanical pulping is thermomechanical pulping (TMP), as illustrated in Figure 1.1 [2].

One method of pulping is known as refining. In the refining process, wood chips measuring approximately 1.9cm in length, 1.9cm in width, and 0.3-0.6cm in thickness, are fed between two metal discs with at least one of them rotating. The wood fibres are developed by the action of the grooves and bars of the plates mounted on the surface of the two discs [1]. TMP involves the presteaming and first-stage refining at an elevated temperature and pressure, followed by a second refining stage under pressure or at atmospheric pressure [1].

After refining, the pulp is fractionated into accepts and rejects streams by the primary screen. The symbols in Figure 1.1 are associated with the screens and represent feed (f), accepts (a), and rejects (r). The accepts stream contains fibres sufficiently refined, and require little subsequent processing before being fed to the paper machine. The rejects stream is fed to a secondary screen. The accepts of the secondary screen are recycled back and combined with the feed to the primary screen. The rejects stream of the secondary screen contain fibres and particles which need further refining. This stream is sent to the rejects refiner.

The output of the reject refiner is sent to the refined rejects screen. The operation of the reject screen is the same as the primary screen. The accepts are combined with the accepts of the primary screen and sent to the paper machine. The reject stream is recycled back to the rejects refiner for further refining.

### **1.1.2 Pressure Screening**

In general, a pulp screen is designed to selectively remove certain constituents from a pulp feed so that the properties of the accepted pulp are more suitable for papermaking. At the same time, it concentrates the objectionable material so that it can be treated or discarded. A schematic diagram of a pulp pressure screen is shown in Figure 1.2 [3]. One of the undesirable particles in a pulp slurry are called shives, bundles of fibres at least *3mm long* and *0.08mm wide* [1]. Figure 1.3 illustrates the construction of a typical pulp screen [4]. The pulp enters the screen at the top of the screen basket and travels towards the bottom where the rejects leave the screen. As the fibres travel through the screening unit, the centrifugal force created by the spinning rotor pushes the fibres towards the screen basket. The fibres that pass through the screen basket leave in the accepts stream, while the fibres filtered by the screen exit through the rejects stream. In some pulp screens, shower water is applied directly to the screen surface, washing the small fibres through the screen.

The screen studied for the project is a Hooper PSV-2600 with XT-type rotors [5]. The rotor has foil blades and the tips are 1.5 inches long, as shown in Figure 1.4. The rotor is the circular object at the center of the diagram. The rotor is constructed by a series of stacked conical pieces as shown in Figure 1.5 [3]. Since the feed stream enters the screen from the top, the flow of the feed is from top to bottom. As the pulp

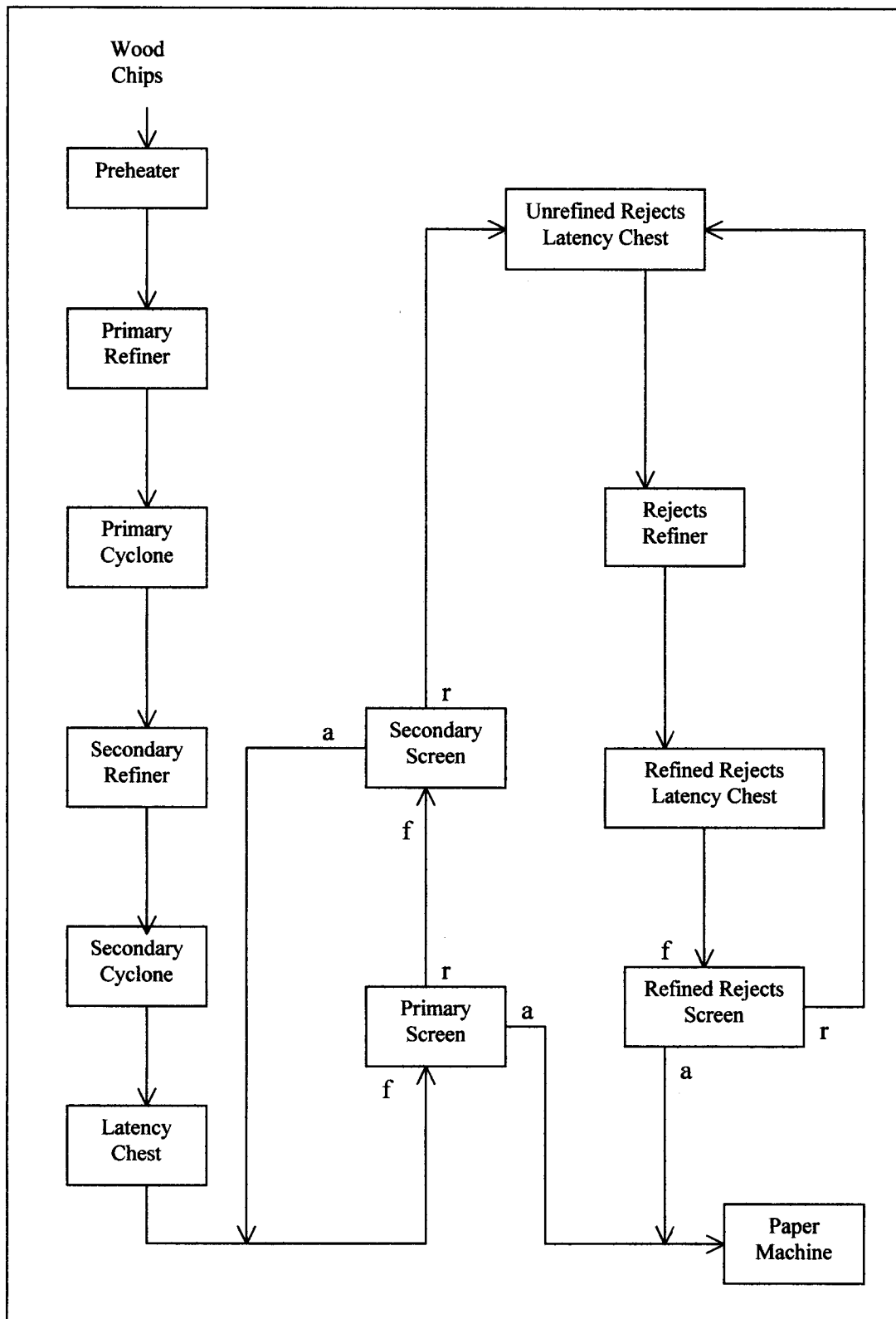


Figure 1.1. A flow diagram of refining and screening stages in a TMP plant.

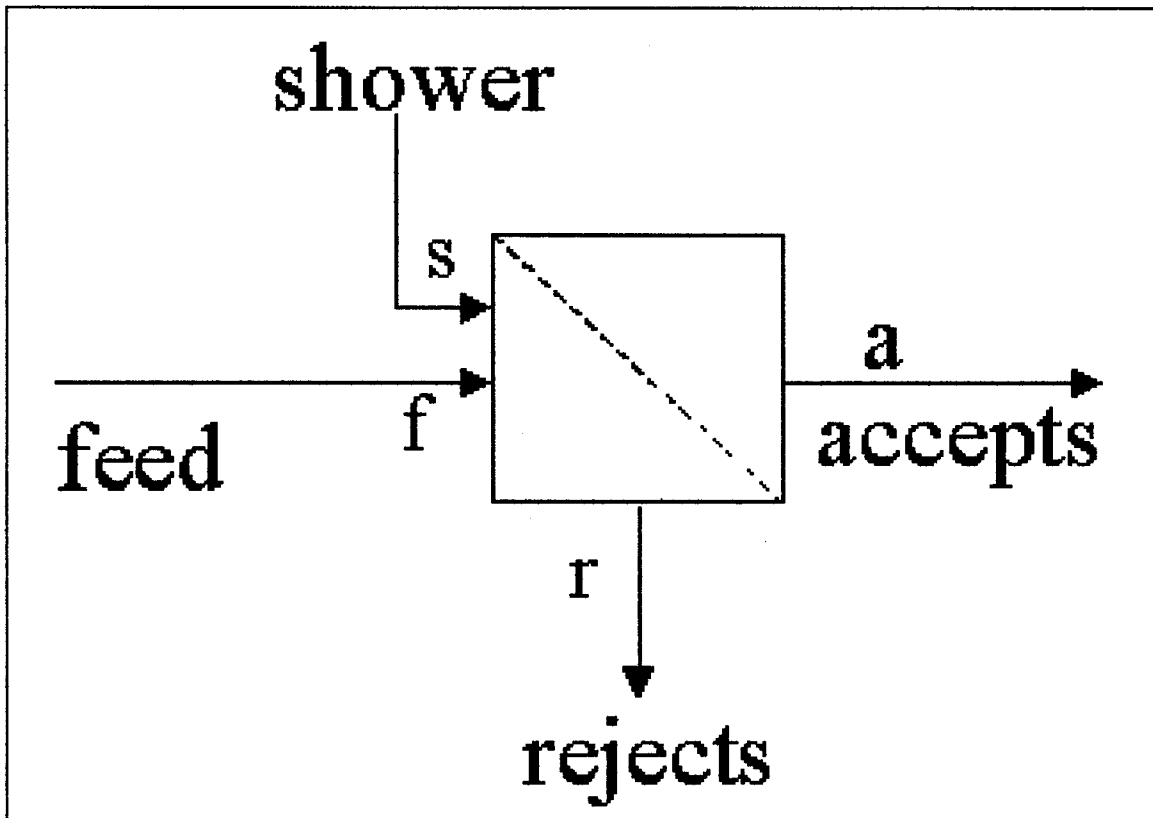


Figure 1.2. Schematic diagram of the rejects pressure screen showing the feed, shower, accepts, and rejects

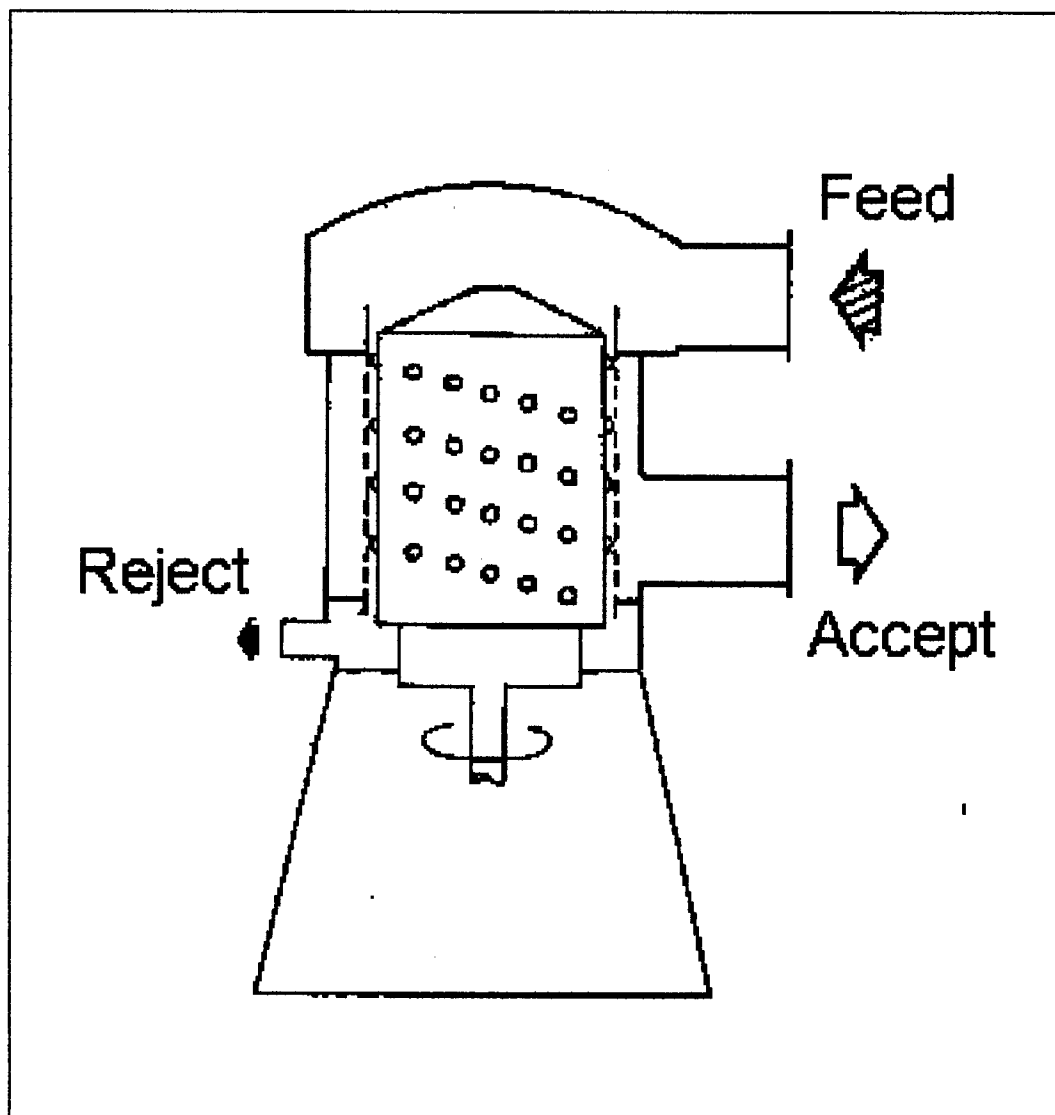


Figure 1.3. A typical pulp screen.

slurry moves downwards, the spinning of the rotor forces the slurry towards the screen basket. The foil blades are shown in Figure 1.5 as the arms extending from the stacked conical pieces. Any fibre passing through the screen exits through the accepts port, while the other fibres continue to the bottom of the screen where it exits through the rejects stream. The screen basket is typically  $3.175\text{mm}$  ( $0.125\text{ inches}$ ) thick, and the hole diameter is  $2.413\text{mm}$  ( $0.095\text{ inches}$ ).

The operation of the refined rejects pressure screen involves two input streams and two output streams. One of the input streams, the feed, is the output of the reject refiner. The feed stream contains a large range of particle sizes. The other input stream is the shower, (or dilution) water. The shower water is required to wash the small fibres into the accepts stream, and prevent the screen from plugging. The flow of shower water must be controlled to prevent the screen from plugging when the shower flow ( $Q_s$ ) is too low, and to prevent excessive dilution when the flow is too high. Excessive dilution causes an increase in energy consumption downstream from the screen to remove the extra water. The flow of the shower water is controlled by a dilution valve.

A mass balance of the screen is given in Equation (1.1). Since the consistency ( $C$ ) of each of the streams is very low (1-3wt%), the density ( $\rho$ ) of each stream is assumed to be equal, and approximately equal to the density of water, as shown in Equation (1.2). Therefore, each of the density terms in Equation (1.1) can be eliminated. The terms associated with the shower stream in Equation (1.1) can also be eliminated because the consistency of the shower water is approximately zero.

$$Q_f C_f \rho_f + Q_s C_s \rho_s = Q_r C_r \rho_r + Q_a C_a \rho_a \quad (1.1)$$

$$\rho_f = \rho_r = \rho_a \cong \rho_{H_2O} \quad (1.2)$$

where  $Q_f, Q_s, Q_r, Q_a$  are the volumetric flow rates of the feed, shower, rejects and accepts streams.  
 $C_f, C_s, C_r, C_a$  are the consistencies of the feed, shower, rejects and accepts streams.  
 $\rho_f, \rho_s, \rho_r, \rho_a$  are the densities of the feed, shower, rejects and accepts streams.  
 $\rho_{H_2O}$  is the density of water.

The screen divides the flow of the pulp stock into two output streams based on the probability that the small fibres passing through the screen is greater than the probability of the large fibres passing through the screen. Ideally, all the small fibers are passed through the screen from the inside to the outside, forming



the accepts stream, while the large fibres are blocked by the screen, and continue into the rejects stream. Since the screening is based on the probability, some of the small fibres go to the rejects stream while some of the large fibres can pass through the screen to the accepts stream. Since the screen hole diameter is *2.413mm (0.095 inches)* and shives are *3mm long* and *0.08mm wide*, they can pass through the screen when oriented lengthwise [6].

The reject stream contains all particles and water not passing through the screen. The reject stream will contain the shives, most of the large fibres, and some of the small fibres. The flow of the reject stream can be controlled by an external valve. Too little reject flow can cause the screen to become plugged by the fibres, while too large a flow may lead to poor fractionation of the feed stream, and an increased load on the rejects refiner.

There are two major parameters which the screen operator can control to affect the efficiency of shive removal and fractionation. They are reject rate and dilution. The reject rate is the main control parameter for the operator [1]. Reject rates can be controlled, and over a reasonably wide range. Dilution is used to keep the screen plate from plugging, and is controlled in proportion to the flow rate of the feed stream.

Although perforation size can affect the fractionation efficiency of the screen, it cannot be controlled or manipulated during daily operations which would require a shutdown of the screen in order to change the screen plate. Feed consistency can also be manipulated by the operating variables of the refiner, but is difficult to alter in an attempt to optimize the operation of the screen.

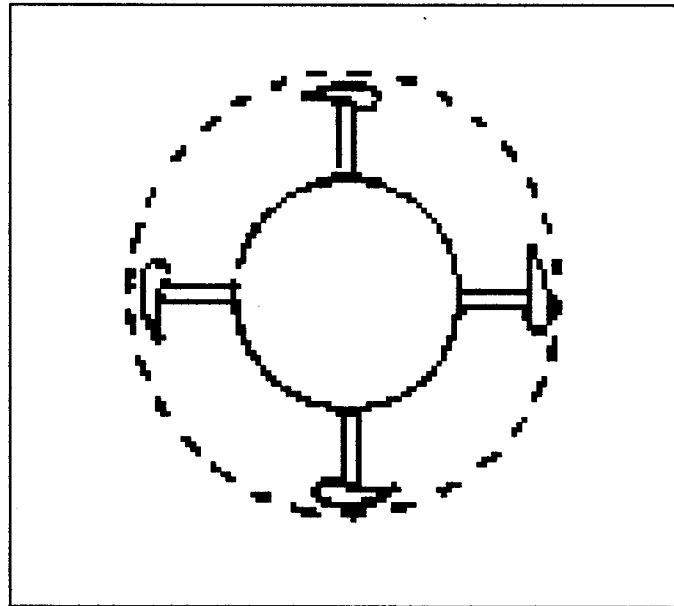


Figure 1.4. The screen rotor and foil blades.

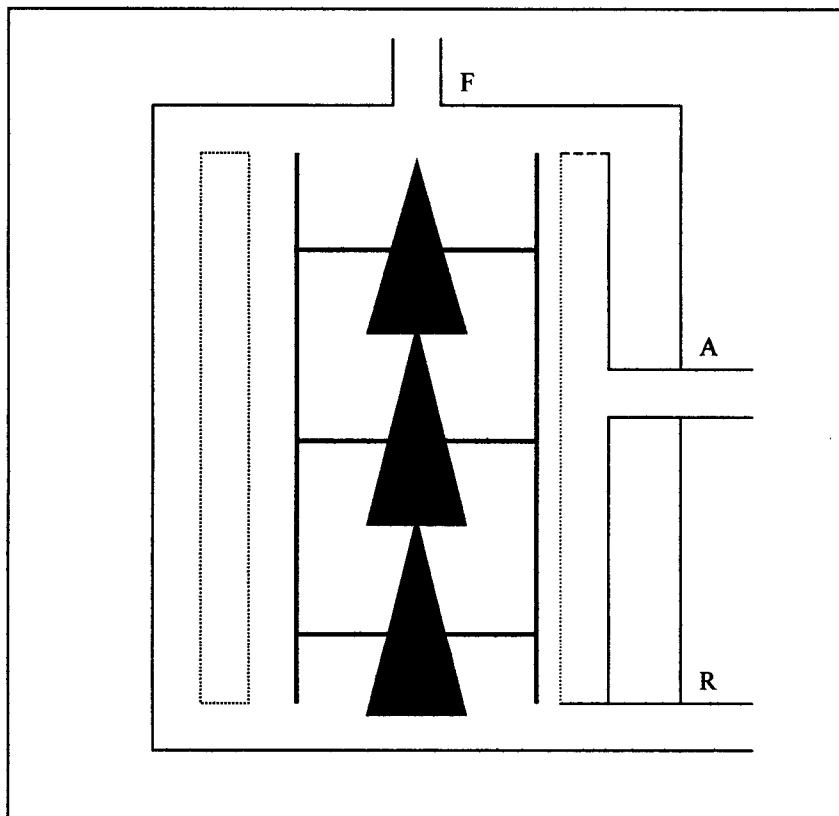


Figure 1.5. Cross-sectional view of the screen rotor design.

The operation of the refined reject pressure screen is controlled by four separate variables: accepts flow rate, accepts differential pressure, shower to feed flow ratio, and volumetric reject ratio. The setpoint values for the shower and reject ratio's requires the knowledge of the feed flow rate. The shower and reject flow transmitters are located on each of the dilution, rejects and accepts streams. The feed flow rate is calculated by a mass balance using measurements of the stock flow of each stream, given in Equation (1.3), assuming all densities to be equal.

$$Q_f = Q_a + Q_r - Q_s \quad (1.3)$$

The accept pressure control valve is used to control the flow through the screen plate and to keep the screen plate from plugging. The type of rotor used is a factor in determining the sign of the differential pressure across the screen. The pressure differential is calculated using Equation (1.4). In some cases, the screen acts as a pump, causing the accepts pressure to be greater than the feed pressure.

$$\Delta p = p_f - p_a \quad (1.4)$$

where  $p_f, p_a$  are the pressures of the feed and accepts streams.

The flow rate through the screen is controlled by adjusting the pressure differential setpoint. If a larger accepts flow rate is desired, the  $\Delta p$  setpoint must be increased. If the  $\Delta p$  is negative this means making the  $\Delta p$  less negative. The increased setpoint value will cause the accepts valve to open. Opening the accepts valve will cause the pressure in the accepts stream to decrease, thereby increasing the  $\Delta p$ . If the  $\Delta p$  is positive, it must be increased for a larger accepts flow rate. Opening the accepts valve will cause the accepts pressure to drop, and the  $\Delta p$  to increase as desired.

If the screen plate begins to plug, the result will be an increased pressure drop because of a low pressure in the accepts stream. To return the differential pressure to its setpoint value, the accepts valve will close, further reducing the flow through the screen. Closing the accepts valve will create a backwash effect on the screen by flushing more of the fibres to the rejects stream and returning the screen to normal operations.

The flow of shower (or dilution) water to the screen is manipulated to control the consistency in the screening zone. Controlling the consistency on the screening zone prevents the screen from plugging

and limits the water in the accepts stream. Shower flow should always be kept to a minimum, only using enough to maintain stable operation because of limited capacity [1].

### 1.1.3 Pressure Screen Models

Mathematical modelling of pulp screens has evolved over the past twenty years. In the early 1980's George Nelson [7] contributed a single number screening performance index. This index is known as the screening quotient,  $Q$ , given in Equation (1.5).

$$Q = \frac{E_c}{E_c + R_w - E_c R_w} \quad (1.5)$$

where  $E_c$  depends on the ratio of the shive concentration in the accepts to the level in the feed, referred to as the cleanliness efficiency of the screen

$R_w$  is the reject rate by weight

The screening quotient offers a quick calculation of the screen performance without being directly tied to the reject rate. The value of the screening quotient varies less than cleanliness efficiency for a change in reject rate. This method allowed for the comparison of screens regardless of the operating reject rate of each screen.

Wakelin [3] suggested the screening quotient was only valid for components that are small proportions of the feed, and that using a volumetric reject rate is a better method of predicting the screen performance.

$$S_F C_F X_{F\varphi} = S_A C_A X_{A\varphi} + S_R C_R X_{R\varphi} \quad (1.6)$$

The fibre removal efficiency, Equation (1.7), is defined as the mass of the fibre length  $\varphi$  in the rejects stream divided by the mass of fibre length  $\varphi$  in the feed stream.

$$E_{R\varphi} = \frac{M_{R\varphi}}{M_{F\varphi}} = \frac{X_{R\varphi} M_R}{X_{F\varphi} M_F} = \frac{X_{R\varphi}}{X_{F\varphi}} R_m \quad (1.7)$$

but volumetric reject rate is defined as

$$R_v = \frac{S_R}{S_F} \quad (1.8)$$

Using the mass balance of the screen given in Equation (1.6) and the definition of the fibre removal efficiency, Wakelin defines a removal efficiency relationship based on the volumetric reject rate as

$$E_{R\varphi} = \frac{R_v}{1 - (1 - p_\varphi)(1 - R_v)} \quad (1.9)$$

$$p_\varphi = \frac{X_{A\varphi} C_A}{X_{R\varphi} C_R} \quad (1.10)$$

where  $E_{R\varphi}$  is the removal efficiency for component  $\varphi$ .

$R_v$  is the volumetric reject rate

$p_\varphi$  is the probability of acceptance

$S_F, S_A, S_R$  are the mass fractions of fibres of component  $\varphi$ , in the feed, accepts and rejects streams.

$X_{F\varphi}, X_{A\varphi}, X_{R\varphi}$  are the mass fractions of fibres of component  $\varphi$ , in the feed, accepts and rejects streams.

$C_F, C_A, C_R$  are the stock consistencies in the accepts and rejects streams.

Wakelin then defines the mass fraction of fibre in the reject stream,  $X_{R\varphi}$ , as a function of the volumetric reject rate in Equation (1.11).

$$X_{R\varphi} = \frac{X_{F\varphi} C_F}{C_R (1 - (1 - p_\varphi)(1 - R_v))} \quad (1.11)$$

where  $X_{F\varphi}$  is the mass fraction of fibres of component  $\varphi$  in the feed stream.

$C_F$  is the consistency of the feed stream.

Wakelin also points out that most screen control is based on the volumetric reject rate rather than the mass reject rate.

Gooding and Kerekes [4,8] proposed that the extent to which fibres pass through individual apertures can be represented by a quantity called "passage ratio." Passage ratio ( $P$ ), shown in Equation (1.12), is defined as the concentration of fibres of length  $l$  passing through a screen aperture ( $c_s$ ), divided by the concentration of fibres immediately upstream of the aperture ( $c_u$ ).

$$P(l) = \frac{c_s(l)}{c_u(l)} \quad (1.12)$$

The passage ratio becomes important when modelling the flow within the screen. An analytical model of the screen can be created using one of two methods. The methods are a mixed flow model or a plug flow model. The following section derives the equations used for both methods of modelling the screen.

#### 1.1.4 Plug-Flow Model

Figure 1.6 shows a schematic diagram of the cross-section between the rotor and the screen plate for a plug-flow model [9,10]. The following derivation follows that of Olson *et.al* (2000). The mass balance equation determined from Figure 1.6 is given in Equation (1.13). The feed stream is represented by  $Qc_u$ , the flow of the pulp stock multiplied by the concentration of fibre in the feed stream. The accepts stream is represented by the  $Pc_u dQ$ . Substituting the passage ratio from Equation (1.12) in  $Pc_u dQ$ , we get  $c_s dQ$ , where  $dQ$  is the difference between the flow of the feed and reject streams ( $Q_f - Q_r$ ).

$$Qc_u = Pc_u dQ + (c_u - dc_u)(Q - dQ) \quad (1.13)$$

where  $Q$  = flow rate in the axial direction

$c_u$  = number concentration of the pulp at point  $u$

By collecting the common terms in Equation (1.13) and assuming the  $dc_u dQ$  term to approximate zero,

Equation (1.14) can be formed.

Rearranging:

$$\frac{dc_u}{c_u} = (P - 1) \frac{dQ}{Q} \quad (1.14)$$

$$\int_{c_r}^{c_f} \frac{dc_u}{c_u} = (P - 1) \int_{Q_r}^{Q_f} \frac{dQ}{Q} \quad (1.15)$$

Both sides of Equation (1.14) can be integrated for the axial length of the screening zone (from feed to reject) as shown in Equation (1.15). After integrating Equation (1.15), the terms can be rearranged to provide Equation (1.16).

$$\frac{c_r}{c_f} = \left( \frac{Q_r}{Q_f} \right)^{P-1} \quad (1.16)$$

In Equation (1.16), the  $Q_r / Q_f$  term can be called the volumetric reject ratio ( $R_v$ ). In the case where shower (or dilution) water is added to the reject screen, the volumetric reject ratio is given by Equation (1.17), and can replace  $Q_r / Q_f$  in Equation (1.16). The result, Equation (1.18) can be used to predict the concentration of fibres in the reject stream ( $c_r(l)$ ) at each fibre length based on the passage ratio, volumetric reject ratio and the feed fibre concentration ( $c_f(l)$ ).

$$R_v = \frac{Q_r}{Q_f + Q_s} \quad (1.17)$$

$$c_r(l) = R_v^{P(l)-1} c_f(l) \quad (1.18)$$

Since  $R_v$  is an operating variable, and  $c_f(l)$  is an input to the model,  $P(l)$  is the only unknown in Equation (1.18). Referring back to Equation (1.16), the left side ( $c_r(l) / c_f(l)$ ) can be described as the reject thickening function,  $t(l)$ , and is shown in Equation (1.19).

$$t(l) = \frac{c_r(l)}{c_f(l)} \quad (1.19)$$

$$P(l) = \frac{\ln t(l)}{\ln R_v} + 1 \quad (1.20)$$

Using experimental trials, values of  $t$  can be determined for all  $l$ . Often values of  $P(l)$  can be calculated from  $t(l)$  and a line fit to the data, in the form of Equation (1.21).

$$P(l) = e^{-l/\lambda} \quad (1.21)$$

where  $\lambda$  is the decay constant.

Once an equation for  $P(l)$  is known, Equation (1.21) can be substituted into Equation (1.18) to predict  $c_r(l)$ .

After the reject fibre concentration is known, the population distribution can be calculated by dividing the concentration at length  $l$  by the sum of the concentrations for all  $l$ , as shown in Equation (1.22).

$$x_{r_p}(l) = \frac{c_r(l)}{\sum c_r(l)} \quad (1.22)$$

A plot of the results obtained by Olson *et. al* is shown in Figure 1.7. Figure 1.7 shows the experimental values for a range of volumetric reject ratios, while the solid line represents the average fibre passage ratio function.

The reject mass flow rate of fibre ( $m_r(l)$ ) at each fibre length can be calculated by multiplying the reject fibre concentration, by the reject volumetric flow rate ( $Q_r$ ), the fibre length ( $l$ ), and the fibre coarseness ( $\sigma$ ), as shown in Equation (1.23). Using the mass balance given in Equation (1.24), the mass flow of the accepts ( $m_a(l)$ ) stream can be calculated. The accepts fibre concentration ( $c_a(l)$ ) can then be calculated from Equation (1.25).

$$m_r(l) = c_r(l)Q_rl\sigma \quad (1.23)$$

$$m_a(l) = m_f(l) - m_r(l) \quad (1.24)$$



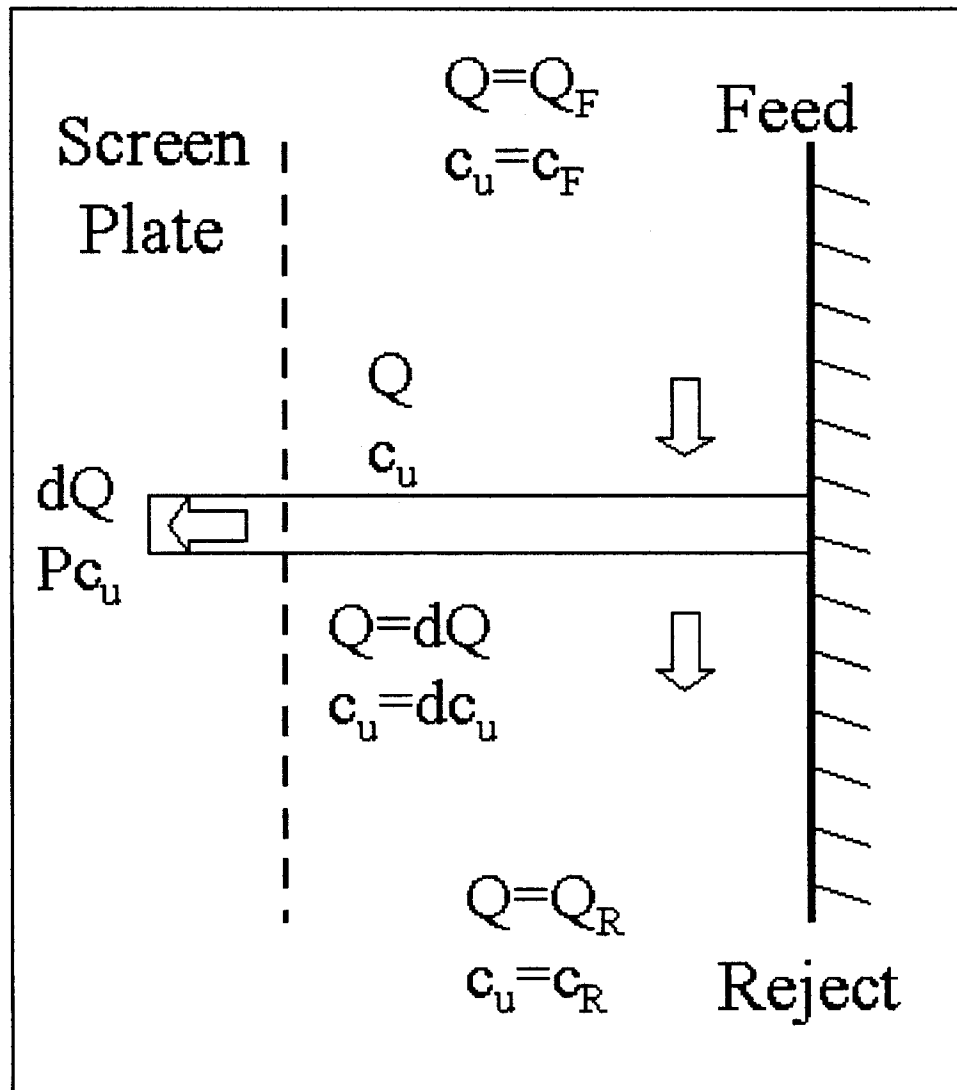


Figure 1.6. Screen cross-section showing the flows through the screen for plug-flow model.

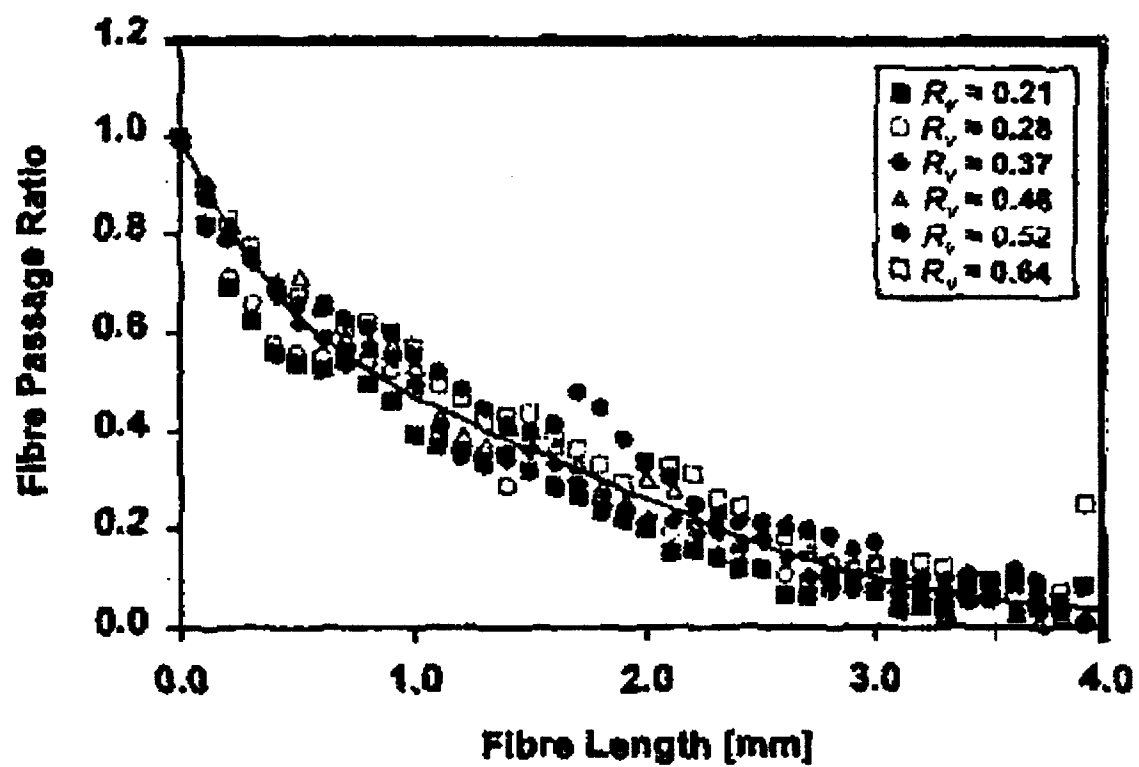


Figure 1.7. Plot of fibre passage ratio from Olson *et. al.*

$$c_a(l) = \frac{m_a(l)}{Q_a \rho_a} \quad (1.25)$$

The population distribution of the accepts stream ( $x_{a_p}(l)$ ) can be calculated by dividing the concentration of fibres at each length by the sum of the concentration of fibres across all fibre lengths, as shown in Equation (1.25).

$$x_{a_p}(l) = \frac{c_a(l)}{\sum c_a(l)} \quad (1.26)$$

The removal efficiency ( $e(l)$ ) of a pulp screen describes its ability to separate the short from the long fibres. It is defined as the mass flow rate of fibres of length  $l$  in the reject stream divided by the mass flow rate of equal length fibres in the feed, as given in Equation (1.27) [9,10,11,12].

$$e(l) = \frac{m_r(l)}{m_f(l)} = R_v \frac{c_r(l)}{c_f(l)} \quad (1.27)$$

$$e_{lg} = \sum_{l=0}^{2mm} e(l) * \Delta l \quad (1.28)$$

$$e_{sh} = \sum_{l=2.05}^{4mm} e(l) * \Delta l \quad (1.29)$$

The short fibre removal efficiency ( $e_{sh}$ ) is the sum for overall efficiency for fibres 0 to 2mm in length as shown in Equation (1.28), while the long fibre removal efficiency ( $e_{lg}$ ) is for fibres 2 to 4m long, given in Equation (1.29). Since very few fibres measure longer than 4mm, they are neglected in calculating fibre removal efficiency.

An unwanted effect of screening is that the pulp consistency in the accept stream is lower than the consistency in the feed stream. If the drop in consistency is large, it can lead to a marked decrease in screen capacity and increased dewatering requirements in downstream processes. Also, the reject consistency will increase, which can lead to high motor loads and plugged reject lines. Normally, the consistency of the reject stream is estimated to be about twice that of the feed stream [9].

The model to predict consistency is a continuation of the pulp fractionation model. The consistency is the sum of the fibre concentrations for all  $l$ , as given in Equation (1.30). This model is applicable for the accepts and the reject streams, but only the reject stream is shown.

$$C_r = \frac{\sum_{l=0}^{7mm} m_r(l)}{\rho_r Q_r} \quad (1.30)$$

The reject consistency is largely dependent on the volumetric reject ratio. A decreasing reject ratio is caused by a decreasing reject flow and/or an increasing shower flow rate. As the reject ratio decreases, the factor between the feed and reject stream consistencies increases. The ratio of the reject consistency to feed consistency is called the reject thickening factor (T), and is given in Equation (1.31). A plot of reject thickening factor versus volumetric reject ratio is shown in Figure 1.8 [11].

$$T = \frac{C_r}{C_f} \quad (1.31)$$

The calculation of the accepts consistency can be done using the same method as the rejects stream once the distribution is known. Figure 1.9 shows a plot comparing the consistencies of the feed, reject and accept streams, versus the volumetric reject ratio. This shows the significance of the reject ratio on the operation of the screen. As the reject ratio decreases, the consistency of the reject stream exponentially increases, and screen plugging becomes a concern.

### 1.1.5 Mixed-Flow Model

When modelling the screen using the mixed-flow method, it is assumed that the concentration of the rejects stream is equal to the concentration in the screening zone for all fibre lengths. A schematic diagram of the mixed-flow, or continuous stirred tank reactor, method of modelling the screen is illustrated in Figure 1.10.

$$c_r(l) = c_u(l) \quad (1.32)$$

Also, the concentration of the accepts stream is assumed constant for each of the screen holes.

$$c_a(l) = c_s(l) \quad (1.33)$$

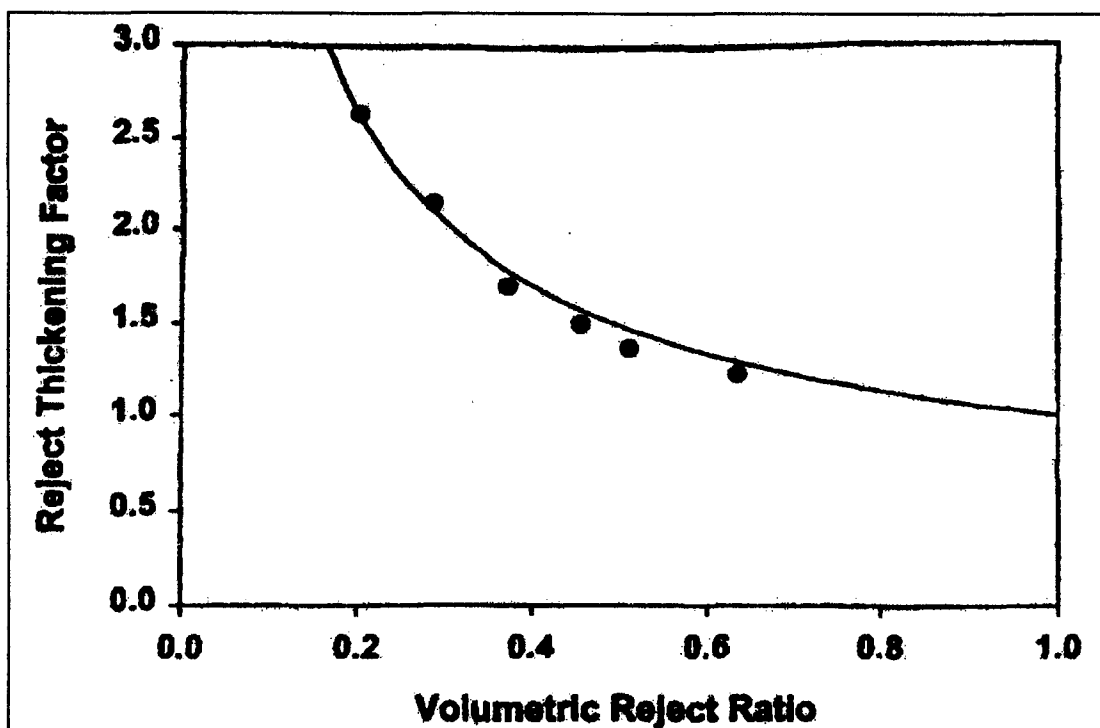


Figure 1.8. Reject fibre thickening factor for the plug-flow method.

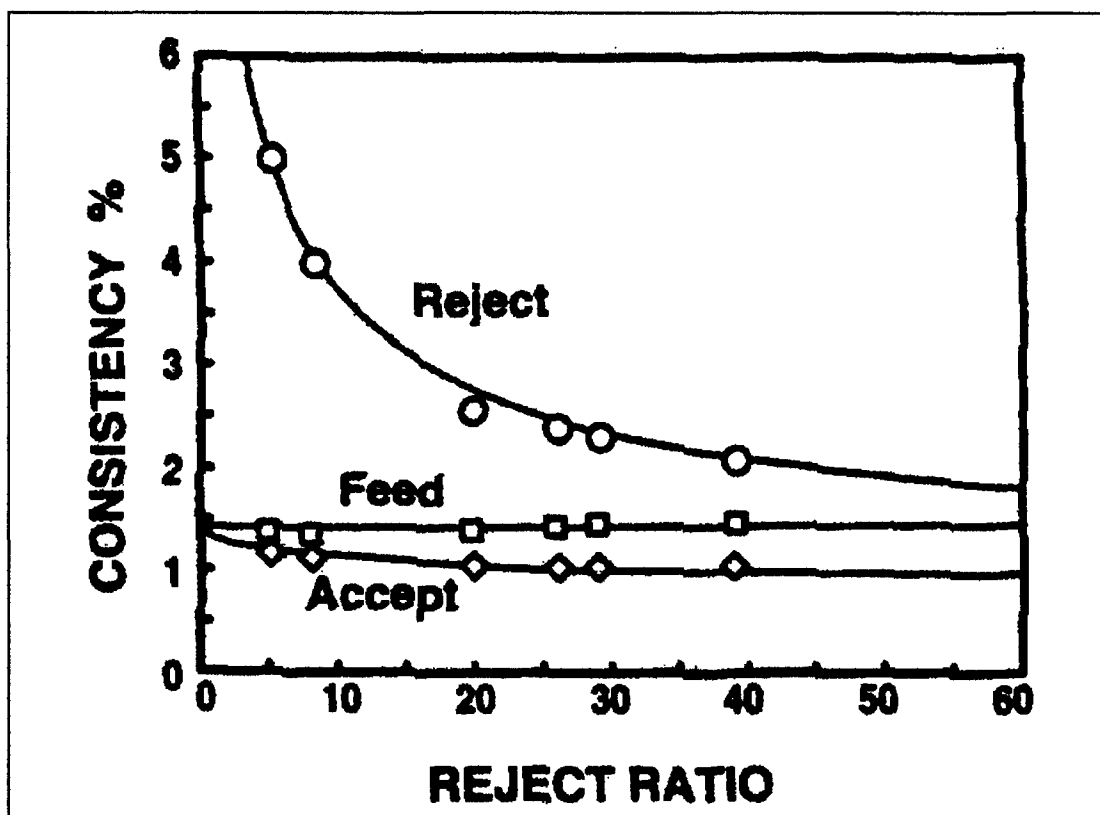


Figure 1.9. Reject consistency increases exponentially as reject ratio decreases.

Equations (1.32) and (1.33) can be substituted into Equation (1.12) to produce

$$c_a(l) = P(l)c_r(l) \quad (1.34)$$

but we want  $c_r$  in terms of  $R_v$ ,  $c_f$  and  $P$ . We can rearrange Equation (1.1) and substitute into the left side of Equation (1.34), to get Equation (1.35).

$$\frac{Q_f c_f(l) - Q_r c_r(l)}{Q_f - Q_r} = P(l)c_r(l) \quad (1.35)$$

and substituting in  $R_v$  from Equation (1.17), we can come up with

$$c_r(l) = \frac{1}{P(l) - P(l)R_v + R_v} c_f(l) \quad (1.36)$$

Equations (1.19) through (1.31) are applicable to the mixed-flow method for predicting the variables for the accept and reject streams, once  $c_r(l)$  is calculated from Equation (1.36).

#### 1.1.6 Freeness Model

The quantity of fines are extremely important to papermaking chemistry and play a decisive role in retention, drainage and paper properties. Most of the literature about retention and drainage behaviour is based on chemical pulp furnishes, and the understanding of mechanical pulps is often an extrapolation of this knowledge. Improvements obtained with retention aids are fairly widely documented in the references, while there is very little literature available on drainage performance [13]. Modelling the freeness in this project requires knowledge of the effect of screening on freeness or how freeness is affected by the fibre size distribution of the pulp. Unfortunately, no literature exists on these topics.

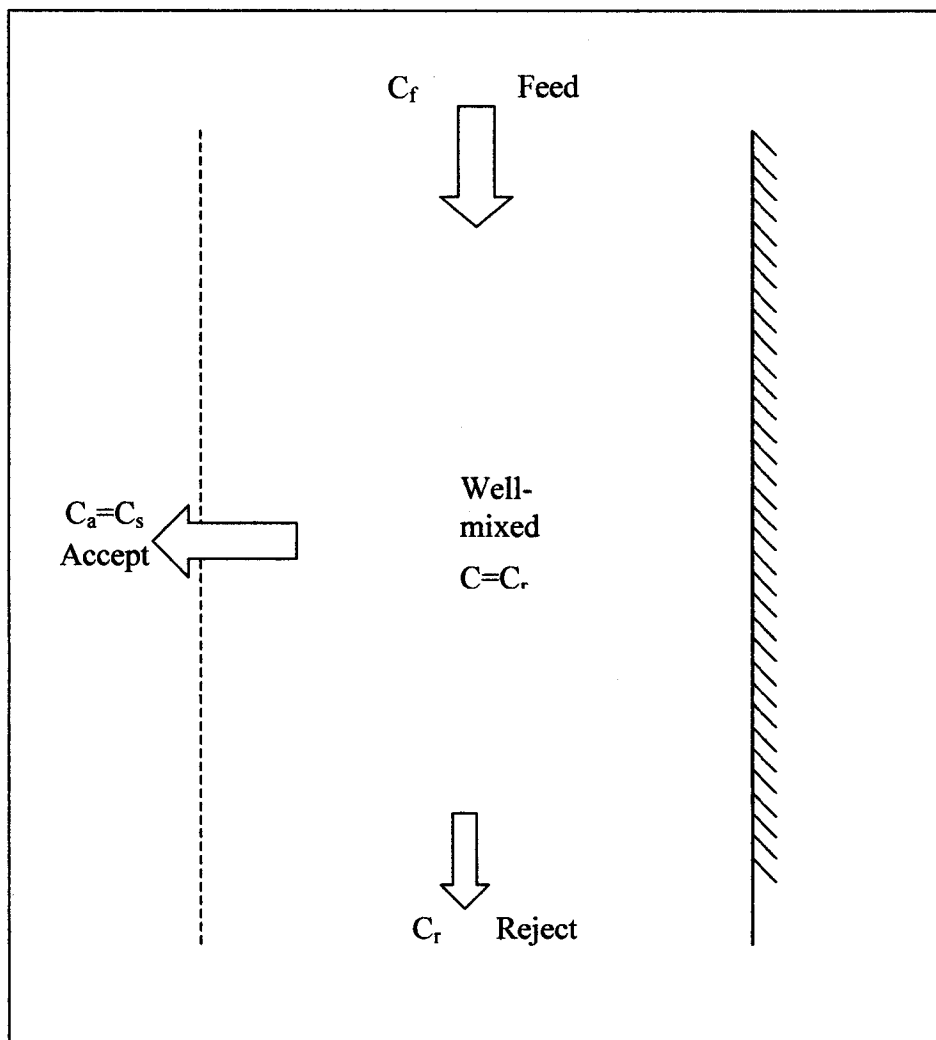


Figure 1.10. Schematic diagram of the mixed flow screen model.

## **1.2 Process Optimization**

Some of the reasons for optimizing the refined rejects pressure screen are the following:

1. Improved fibre separation
2. Higher product quality
3. More efficient use of operators

From these three benefits of optimization others can be found. An example of another benefit is a reduction in energy requirements as a result of the improved fibre separation.

Optimization problems contains three essential elements: an objective function, a model of the process, and equality and inequality constraints. The formulation of objective functions is one of the crucial steps in the application of optimization to a practical problem [14]. The objective function is a mathematical equation describing the cost function of the problem. The cost function contains variables that can be computed from a process model

Equality and inequality constraints must be considered in formulating the objective function. The inequality constraints are the more common of the two. A set of inequality constraints create a bounded region in which the optimum must lie, where as equality constraints creates a line in which the optimum must fall. An example of an inequality constraint is the maximum reject consistency in order to prevent the screen from plugging.

Using the theory presented earlier in the chapter the optimum could be found using a model-based optimization technique. A direct-search method was chosen to allow the control structure of the modified simplex method the power to search for an on-line optimum for any process. The model of the refined rejects pressure screen was used to simulate the ability of the modified simplex method.

### **1.2.1 Sequential Simplex Method**

In 1962, Spendley et al [14] formulated the sequential simplex method (SSM). This direct search method does not require the use of derivatives to determine the search direction, and it has the advantage of being simple to understand and execute [14].



The technique uses a regular geometric figure, the simplex, to find the optimum location of the objective function [14]. Experiments are located such that the objective function is evaluated at the points formed by the vertices of the simplex. One vertex is then rejected as being inferior in value to the others. The general direction of search may then be taken in a direction away from this worst point, being chosen so that the movement passes through the center of gravity of the remaining points [15].

The number of sides on the simplex is  $n+1$ , where  $n$  is the number of manipulated variables in the problem. The length of one side of the simplex is represented by  $m$ . If the number of manipulated variables is two, the method requires an equilateral triangle to find the optimum. From an initial point  $(x_1, y_1)$  and a chosen side length, the location of the remaining points of the simplex can be calculated. In an industrial setting, the starting simplex can be calculated based on the current operating conditions, such as the volumetric reject ratio and the shower to feed ratio. The current conditions provide the initial point  $(x_1, y_1)$ , while the second and third points are calculated using Equations (1.37) through (1.40) [12].

$$a = \frac{m}{n\sqrt{2}} (\sqrt{n+1} + n - 1) \quad (1.37)$$

$$b = \frac{m}{n\sqrt{2}} (\sqrt{n+1} - 1) \quad (1.38)$$

$$(x_2, y_2) = (x_1 + \varepsilon_x * a, y_1 + \varepsilon_y * b) \quad (1.39)$$

$$(x_3, y_3) = (x_1 + \varepsilon_x * b, y_1 + \varepsilon_y * a) \quad (1.40)$$

$\varepsilon_x$  and  $\varepsilon_y$  are either -1 or +1 depending on the location of the initial point. If the initial point is greater than the midpoint between the upper and lower reject ratio limits, then  $\varepsilon_x = -1$ , otherwise  $\varepsilon_x = 1$ . This procedure ensures that no simplex can be calculated with points outside the selected boundary. The same procedure is used to determine the value of  $\varepsilon_y$ .

Once the second point of the simplex has been calculated, the process is adjusted to correspond with the second point. The objective function is then evaluated after steady state has been achieved, and the

measurements from the process are retrieved. The same procedure is repeated for the third point of the starting simplex.

In moving toward the optimum operating conditions, two basic operations can be performed on the simplex. The basic simplex operations are reflection and contraction. Figure 1.11 shows the flow diagram of the sequential simplex method.

The SSM must begin by calculating a simplex based on one point. If attempting to optimize two variables of a process, the simplex will contain three points. Each of three points in the simplex will be given a row in a matrix labelled  $P$ , and evaluated using the performance equation and ranked from lowest to highest. The dimensions of the  $P$  matrix will be  $n+1$  rows by  $n$  columns. The lowest (or best point) will be labelled with the subscript  $ilo$ , the middle point labelled  $inhi$ , and the highest (or worst point) labelled  $ihi$ . When searching for a minimum, the highest point is discarded, and the simplex is reflected away from it. The newest point must now be evaluated and ranked against the two remaining points. Each point is evaluated using the performance equation to determine which is furthest from the optimum. If point 1 is determined to be the worst, the simplex is then reflected about the line drawn between points 2 and 3, as shown in Figure 1.12. Point (1) is discarded and replaced by point (4), calculated using Equation (2.3) [15]. This creates the new simplex, with vertices at points (2-3-4).

$$P_{new,i} = \sum_{j=1}^{n+1} P_{j,i} - 2 * P_{ihi,i} \quad (\text{for } i=1 \text{ to } n) \quad (1.41)$$

where  $P_{new}$  is the newest point calculated in the simplex.

$P_{ihi,i}$  is the value of the highest (or worst) point in the simplex.

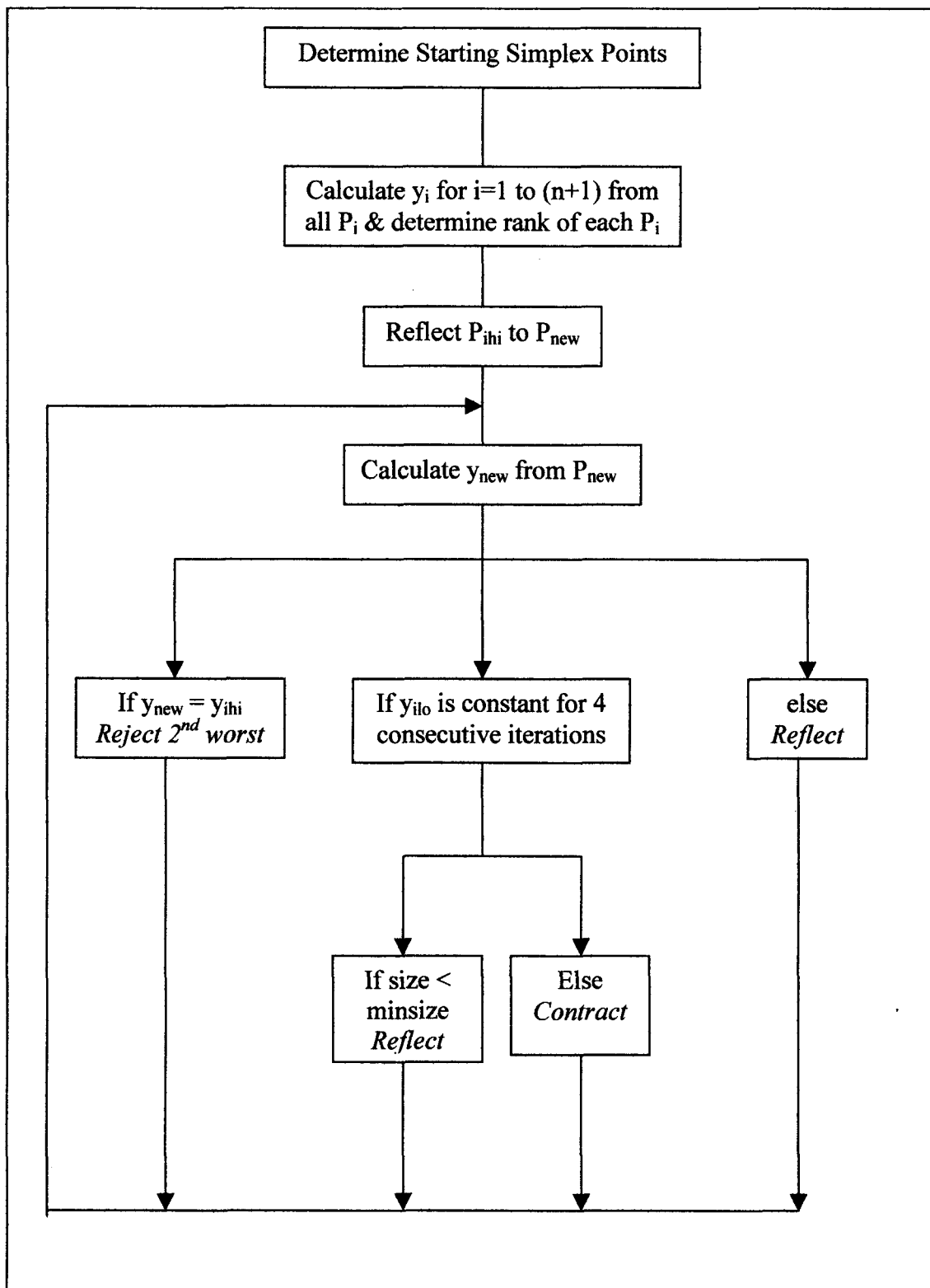


Figure 1.11. Flow diagram of the sequential simplex method.

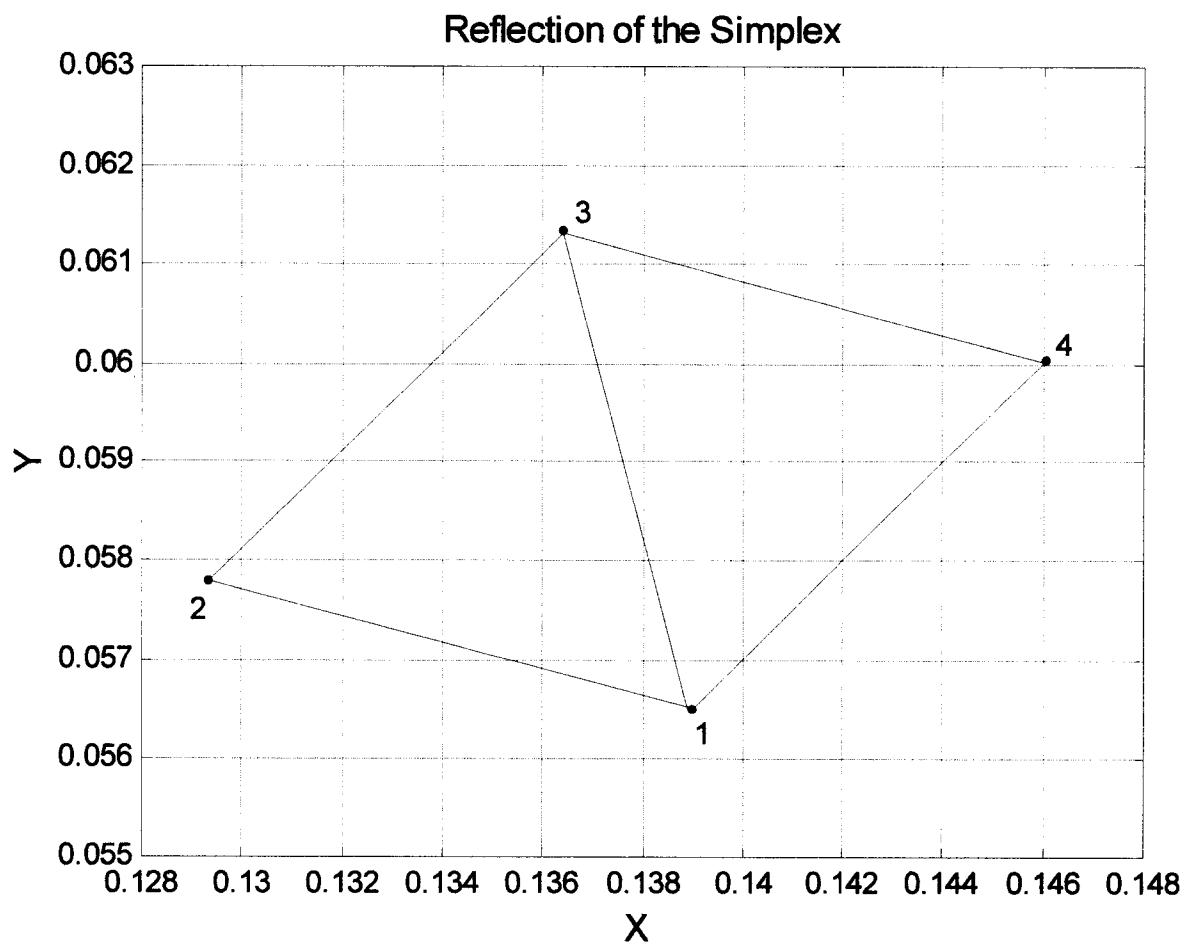


Figure 1.12: Reflection of a simplex away from worst point.

In Figure 1.12,  $(P_{ih1}, P_{ih2}) = (x_1, y_1)$ , and  $(x_4, y_4) = (P_{new,1}, P_{new,2})$ . The position of the valves are calculated from the ratio's and adjusted to correspond to  $(x_4, y_4)$ . The performance function is then evaluated using the values of point (4) to determine the next move of the simplex.

The procedure of reflecting the simplex away from the worst point will continue as long as the newest point in the simplex is not the worst. If the newest point is worse than the remaining two points, the simplex is either straddling a ridge or stuck in a valley. If this occurs the simplex will simply reflect back and forth between two points. This can be corrected by not allowing the simplex to return to a point that it just left. Instead of rejecting the worst point, the second worst point ( $P_{mhi}$ ) in the simplex is rejected, using Equation (1.42) [15]. This procedure can slow the rate that the simplex moves toward the optimum, but helps ensure that when the optimum region is found, it isn't a local minimum.

$$P_{new,i} = \sum_{j=1}^{n+1} P_{j,i} - 2 * P_{mhi,i} \quad (\text{for } i=1 \text{ to } n) \quad (1.42)$$

As the simplex approaches the optimum location, the simplex will begin to pivot around one point (3), as shown in Figure 1.13. The only method of getting closer to the optimum location is to reduce the size of the simplex. A reduction in simplex size should occur if the best-valued vertex remains unchanged for more than M iterations, calculated using Equation (1.43) [15]. For example, if  $n = 2$ , then the size of the simplex should be reduced when the best valued point remains unchanged for more than 3.5 movements of the simplex.

$$M = 1.65 * n + 0.05 * n^2 \quad (1.43)$$

In Figure 1.13, point (3) is the best point for four consecutive iterations, therefore, the simplex should be reduced in size. The simplex size is reduced by holding one point constant, the best point, and recalculating the other two, creating a new simplex. The two new points will be calculated using the procedure for calculating the starting simplex except the value of  $m$  is decreased by a chosen factor. The simplex in Figure 1.13 is shrunk from (3-5-6) to the new simplex (3-7-8).

The procedure of cycling about a single point will continue and the size of the simplex will be further reduced, until a specified minimum size is achieved. Once the simplex has reached the minimum

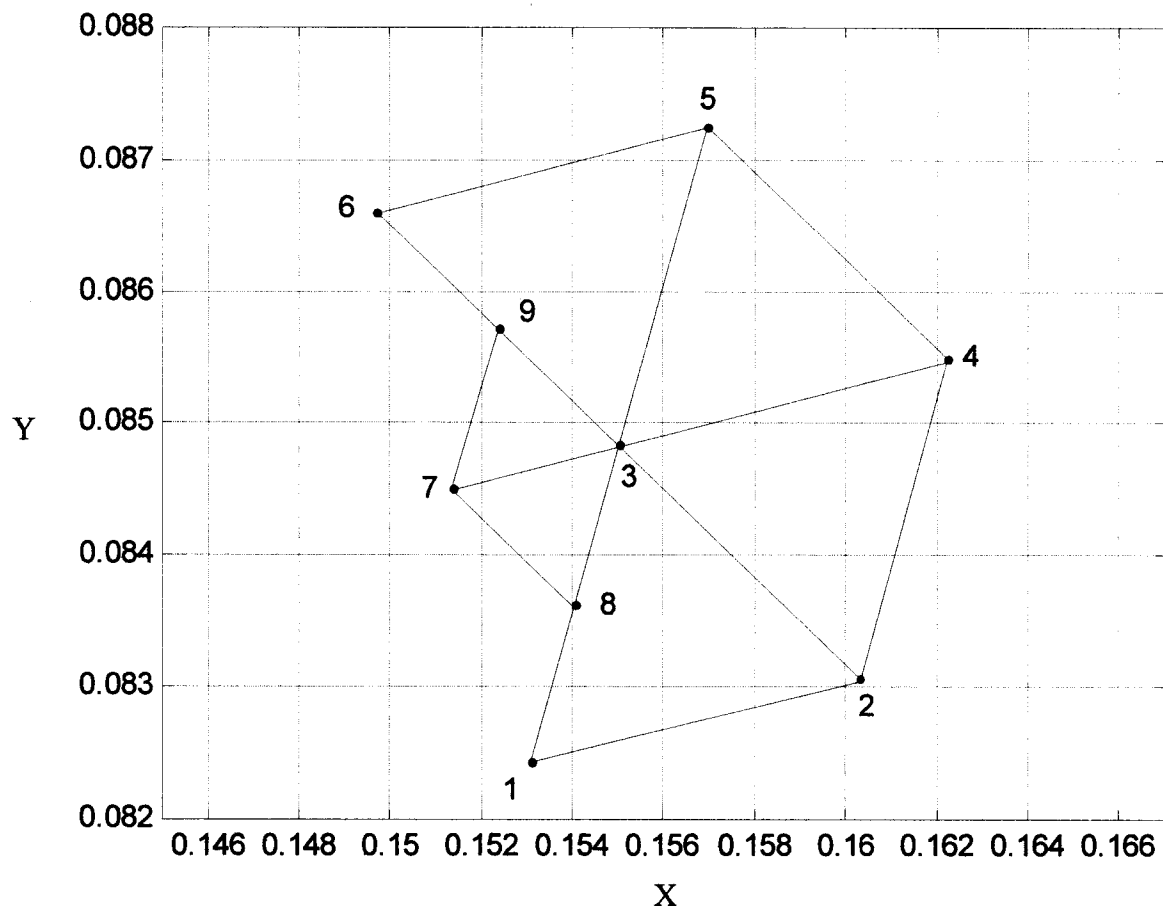


Figure 1.13: Contraction of equilateral simplex towards optimum location.

size, the simplex will continue to reflect away from the worst point and begin to cycle until interrupted by some other incident.

### 1.2.2 Nelder and Mead Method

The Nelder and Mead method is an extension of the sequential simplex method. Nelder and Mead suggested that the size of the simplex could be expanded, allowing the simplex to accelerate towards the optimum. In the Nelder and Mead method, the simplex adapts itself to the local landscape, elongating down long inclined planes, changing direction on encountering a valley on an angle, and contracting in the neighbourhood of a minimum [16].

At each stage of the optimization, one of the simplex points is replaced using one of three operations: reflection, contraction, and expansion. Reflection occurs when the newest point in the simplex is neither the best or worst point when evaluated by the performance equation. Expansion takes place when the newest point in the simplex becomes the best point, while the simplex is contracted when the newest point becomes the worst point in the simplex, but is better than the point rejected from the previous simplex [16]. The Nelder and Mead method is very similar to the sequential simplex method, except during contraction or expansion of the simplex. The Nelder and Mead method only requires one of the points to be replaced. This causes the shape of the simplex to change from the original equilateral shape. The simplex will become elongated when accelerating towards the optimum. A flow diagram of the Nelder & Mead method is shown in Figure 1.14.

The first step of the Nelder and Mead method is to calculate the second and third points of the starting simplex based on the selection of the first point. Once these three points are known, the performance equation must be evaluated at each of the three points. The points of the simplex are labelled best point,  $y_{ilo}$ , middle point,  $y_{inhi}$ , and worst point,  $y_{ihi}$ . The simplex is then reflected away from the worst point using Equation (1.44), to begin the process of searching for the optimum.

$$P_{new} = (1 + \alpha)P_c - \alpha P_{ihi} \quad (1.44)$$

where  $P_{new}$  is the location of the reflected point in the predicted simplex.

$\alpha$  is a positive constant, called the reflection coefficient.

$P_c$  is the location of the centroid of the current simplex.

$P_{ihi}$  is the location of the worst point in the current simplex.

The value of the newest point,  $y_{new}$ , is determined from  $P_{new}$ . The value of  $y_{new}$  is then compared with  $y_{ilo}$  and  $y_{ihi}$  to determine the next simplex operation. If the value of  $y_{new}$  falls between the best point and the middle point, the simplex is reflected again. If the value of  $y_{new}$  is less than the value of  $y_{ilo}$ , the simplex is expanded using Equation (1.45).

$$P^* = \gamma P^* + (1 - \gamma) P_c \quad (1.45)$$

where  $\gamma$  is the expansion coefficient, and is determined using Equation (1.46).

$$\gamma = \frac{dist[P^* P_c]}{dist[P_{new} P_c]} \quad (1.46)$$

If the value of  $y_{new}$  is worse than all  $y$  except  $y_{ihi}$ , the simplex is contracted using Equation (1.47)

$$P^* = \beta P_{ihi} + (1 - \beta) P_c \quad (1.47)$$

where  $\beta$  is the contraction coefficient calculated from the ratio of the distance  $[P^* P_c]$  to  $[P_{new} P_c]$ .

If the value of  $y_{new}$  is worse than all  $y$ , the simplex is contracted using Equation (1.48).

$$P^* = \beta P_{new} + (1 - \beta) P_c \quad (1.48)$$

### 1.2.3 Dynamic Simplex Method

Xiong and Jutan[19] extended the Nelder and Mead simplex method to allow tracking of moving optimum. Xiong and Jutan recognized that in a process that keeps moving continuously, the measurements at a certain point may be valid for only a certain period of time and an older measurement can mislead the tracking algorithm. To overcome this problem, Xiong and Jutan re-measured the best point at the end of each iteration and re-calculated the new direction to allow better tracking.

They also realized that a variable-sized simplex can be susceptible to becoming over elongated in one direction. This can become a problem if the direction of a new optimum is perpendicular to the direction of the elongated simplex. to resolve this potential problem, Xiong and Jutan proposed to fix the simplex size throughout the optimization process.



#### **1.2.4 Modifications required for direct control applications**

The sequential simplex method was created with intention of searching for a stationary optimum. This means that only three simplex operations are required: reflect, reject 2<sup>nd</sup> worst and contract. This becomes a problem when optimizing a process with a moving optimum, such as the refined rejects pressure screen. Optimizing a process with a moving optimum requires an expansion operation to efficiently track the moving optimum. The Nelder and Mead method includes the expansion term and can search out an optimum that is mobile. The Nelder and Mead also changes shape, which can be an advantage, but the simplex could become too elongated. If the simplex becomes too elongated, the simplex moves too slowly in one direction, and too quickly in the perpendicular direction. Another reason for creating a modified simplex is the response to a change in the location of the optimum without re-measuring the best point as is the case when using the Xiong and Jutan Dynamic Simplex method. Although the Dynamic Simplex method may be safe from expanding or contracting in the wrong direction, it may also be slow in responding to large changes in the location of an optimum.

### **1.3 Pressure Screen Optimization**

Allison and Olson [9] also attempted to optimize a screening system. They attempted single and multiple stage screening systems. Allison and Olson used quadratic programming to find the minimum of a multivariable equation. They used equations to model the screen and found the optimum based on the model. Unfortunately, any inaccuracies in the modelling will cause the optimum to be incorrect.

The method used for this project doesn't rely on a model to determine the optimum operating conditions. The only requirements are a cost equation made up of the expectations and constraints, and accurate measurements of the pulp. This method attempts to optimize the screen on-line, whereas Allison and Olson's method was never intended to be on-line.

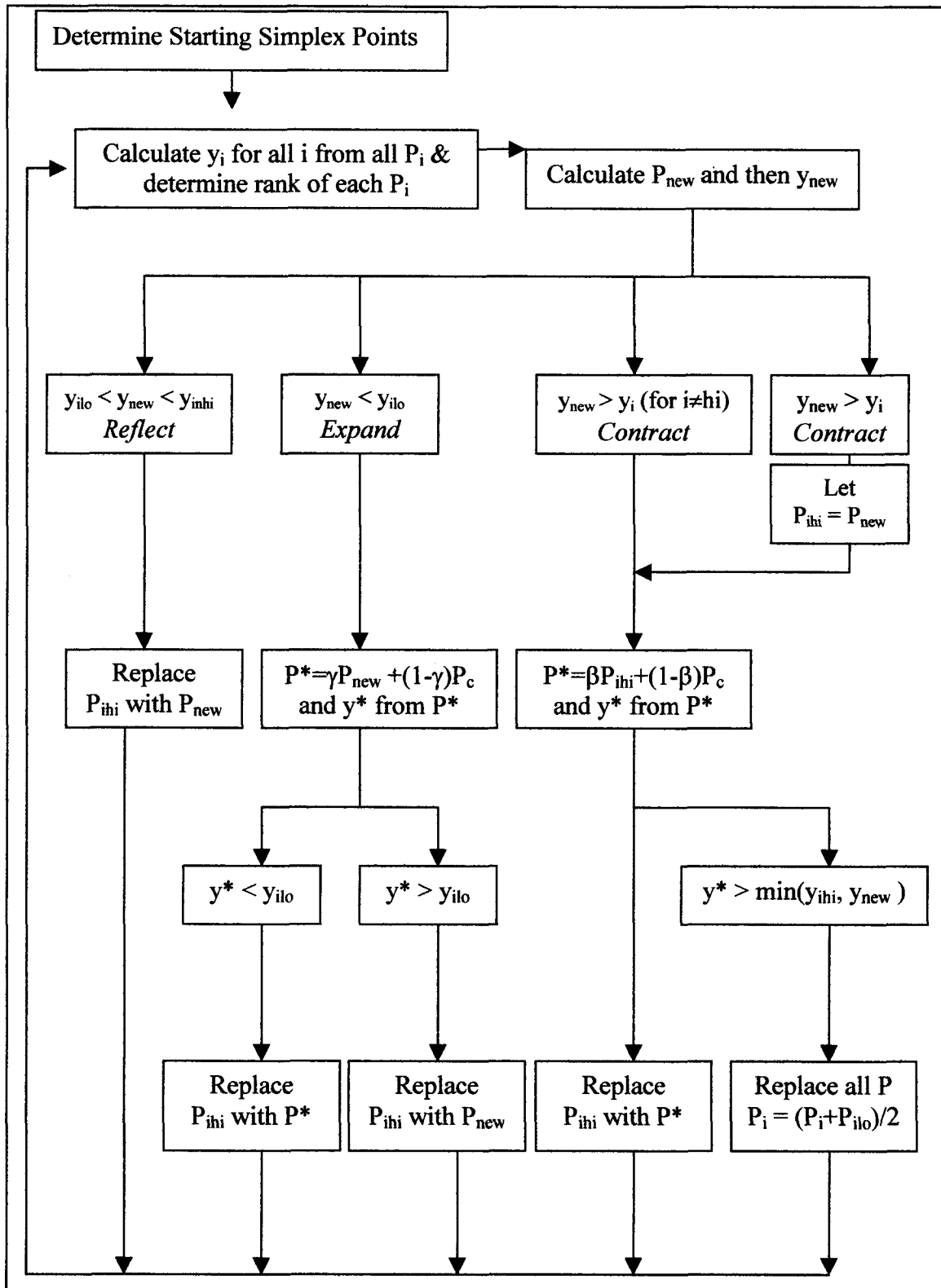


Figure 1.14: Flow diagram of the Nelder & Mead method of optimization.

## 2. The Modified Simplex Methods

The sequential simplex method wasn't created with the idea that the optimum location may wander. If the optimum moves during a time when the simplex is cycling around the best point, the simplex won't realize a change has occurred. The single or best point will continue to be the best point, because the shift in the direction of the optimum will increase the value of the cost equation in the vicinity of the current simplex. Since the cost equation has increased, the best point will be incorrect. The cost of the best point must be re-evaluated to determine whether it is still the best point. In order for the sequential simplex method to search out a moving optimum, another simplex operation must be added.

### 2.1 The Non-equilateral Optimum Following Simplex

The non-equilateral optimum following simplex (NOFS) method uses a combination of the sequential simplex method and the Nelder and Mead method. The NOFS version uses most of the rules of the sequential simplex method, with some of the ideas of the Nelder and Mead method, and some rule changes for contraction. The procedure for the NOFS method is illustrated in Figure 2.1.

The optimization begins the same way as the other methods where the current operating conditions become the first point in the simplex. The performance equation is then evaluated to determine the cost of the current point. The second and third points are calculated using Equations (1.37-1.40). Once the second point is calculated, the operating conditions are adjusted accordingly, and the performance equation is evaluated once the measured values are available. The steps for the second point are repeated for the third to create two matrices, one containing the three points, and the other with the corresponding cost for each point. Each of the three points is ranked based on their cost. The worst is labelled, *ih*, the middle point labelled, *in*, and the best point labelled, *il*. The simplex is initially reflected to replace the worst point using Equation (1.41). After the simplex is reflected, the plant operating conditions are adjusted accordingly, and the cost is determined for the newest point, and each point is ranked again.

Expansion occurs when the newest point in the simplex becomes the best point, and all of the following conditions are fulfilled:

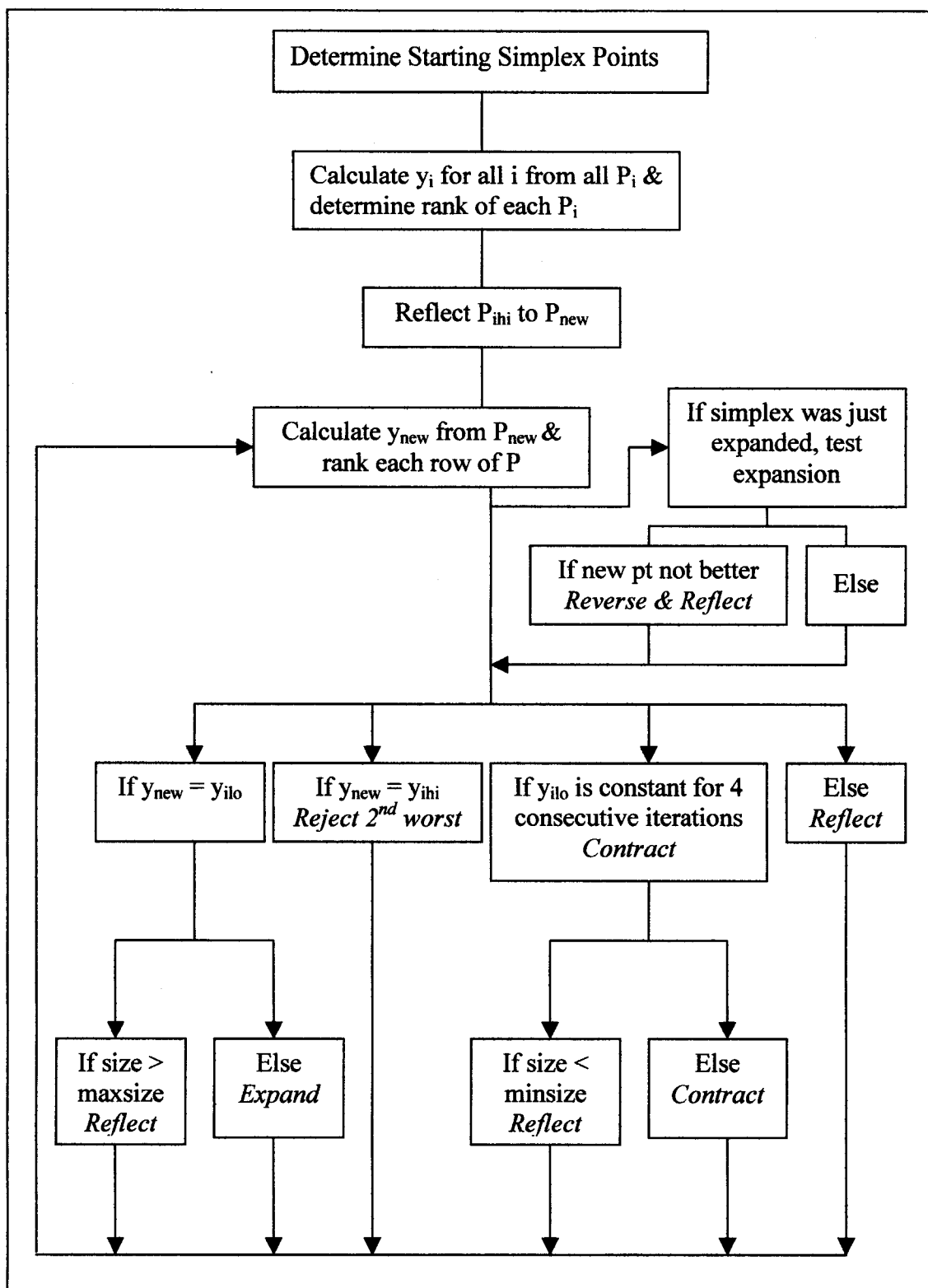


Figure 2.1. Flow diagram of the NOFS method.

1. Maximum simplex size has not been exceeded,
2. The ratio of the distance (centroid to *ilo*) to the distance (centroid to *inhi*), is less than 1.7.
3. The ratio of the distance (centroid to *ilo*) to the distance (centroid to *ihi*), is less than 1.7.

Figure 2.2 shows a simplex that has already been elongated where the distance from the centroid to each point is not equal. The centroid is marked where the three dotted lines meet inside the simplex. The length of line *y* is shorter than both *x* and *z*, therefore as long as the maximum size of the simplex has not been exceeded, this simplex would expand. The size of the simplex is considered the longest line from any simplex point to the centroid. If the maximum simplex size has already been achieved, or any of the three conditions are not met, the simplex will not expand, it will simply reflect away from the worst point.

Expansion of the NOFS method uses the rule from the Nelder & Mead method and Equation (2.1), adapted from Equation (1.41).

$$P_{new,i} = \left[ 5 * P_{ilo,i} - \sum_{j=1}^{n+1} P_{j,i} \right] / 2 \quad (for \ i = 1 \ to \ n) \quad (2.1)$$

Figure 2.3 illustrates how the simplex can become elongated as it accelerates towards the optimum. After the initial simplex (1-2-3) is reflected twice to create simplex (1-4-5), point (5) becomes the best point in the simplex (1-4-5). The simplex is then expanded using Equation (2.1) to create the new simplex (1-4-6).

In the step following a expansion of the simplex, the expansion operation is tested to ensure that the newest point is better than the point replaced. If the cost of the newest point is better than the previous one, the simplex will continue to follow the normal procedure of ranking each point in the simplex and determining the next move. Otherwise, If the cost of the newest point as a result of expansion is worse than the point replaced, the expansion is reversed, and the simplex is reflected by replacing the worst point all in one step. When the simplex is expanded, the point being replaced is placed into a holding variable, and can be recalled if the expansion should be reversed. This point being held replaces the expanded point to recreate the pre-expansion simplex. In order to help keep the simplex moving, the simplex is reflected away from its worst point.

If the newest point in the simplex becomes the worst point, the natural progression of the simplex is to replace the newest point, and return to the previous point. This procedure would cause the simplex to flip back and forth between two points. The optimum location may be located between these two points causing

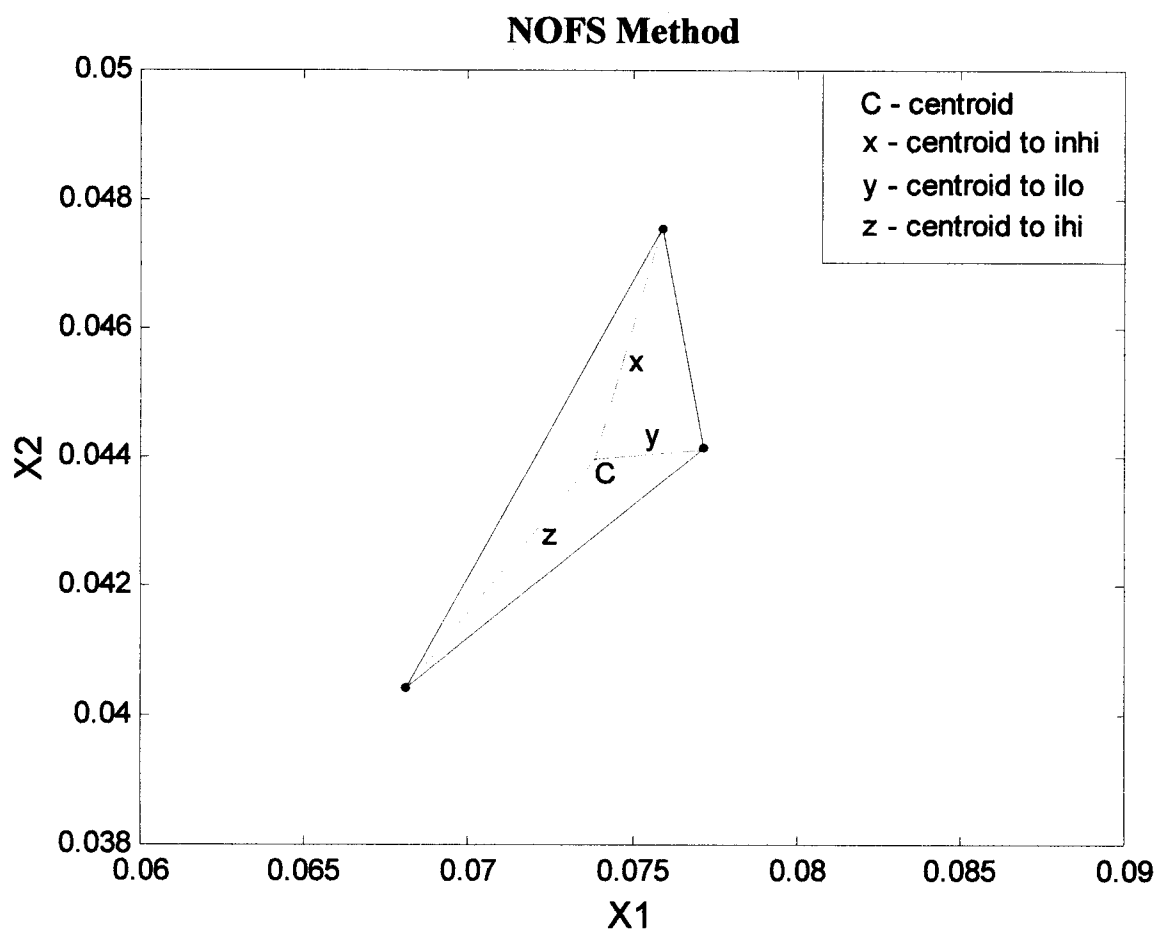


Figure 2.2. Elongated simplex illustrating the location of the centroid.

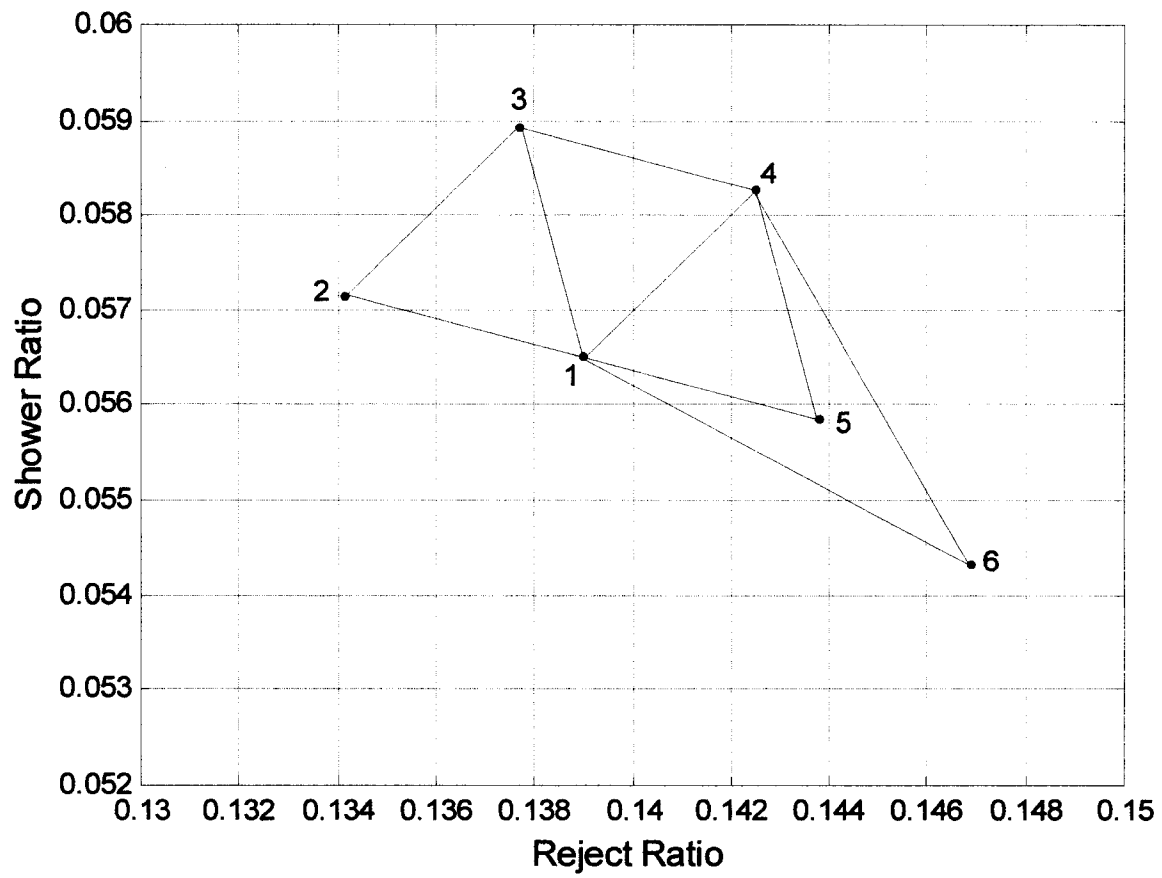


Figure 2.3. Expansion of the simplex using the NOFS method.

the back and forth action, or the simplex may be straddling a local optimum. In order to prevent the simplex from getting caught flipping back and forth, the second worst point is replaced when the newest point is the worst point, using Equation (1.42).

In the NOFS method, the rule for contraction is the same as the sequential simplex method, except the point to be replaced is different. The simplex will be reduced in size if the best point remains the best point for more than  $M$  iterations, based Equation (1.43), and all of the following conditions are met:

1. Simplex size has not been shrunk below the minimum size,
2. The ratio of the distance (centroid to *inhi*) to the distance (centroid to *ilo*), is less than 1.7,
3. The ratio of the distance (centroid to *ihl*) to the distance (centroid to *ilo*), is less than 1.7.

If the simplex size has already been reduced to a selected minimum, or any of the conditions are not met, the simplex is reflected away from the worst point, causing the simplex to continuously pivot about the best point.

Previous simplex methods of optimization have been designed for a static performance equation. The contraction of the simplex about the best point is the obvious procedure for closing in on the optimum location. Normally the decrease in simplex size would be accomplished by replacing the worst point, and calculating a new point closer to the centroid based on Equation (2.2). Figure 2.4 illustrates the procedure of eliminating the worst point during the process of contracting the simplex. After the simplex pivots point (1) four times, point (5) is determined to be the worst in the simplex (1-5-6). Point (5) is then eliminated and point (7) is calculated from Equation (2.2)

$$P_{new,i} = 0.25 * \left( \sum_{j=1}^{n+1} P_{j,i} + P_{ih,i} \right) \quad (for i = 1 to n) \quad (2.2)$$

In an industrial setting, it is possible that the plant will change causing a relocation of the optimum. If the plant does change, previous simplex optimization techniques may take too long to expand to follow a moving optimum, particularly if the simplex has shrunk several times from its original size. For the refined rejects pressure screen the optimum operating conditions will change on a regular basis due to variations in the input stream to the screen. The previous simplex procedure would not realize the value of the performance equation for the best point in the simplex, has probably changed. This could lead to the simplex infinitely pivoting a single point, because it is believed that point is still the best point. This



procedure would result in the cost of the other two points being updated regularly, and the cost of the best point to be incorrect. To prevent this problem, the best point would need to be reevaluated using the performance equation. In a plant this would mean setting the operating variables at the best values and sampling results and determining performance again but this may be inefficient if the optimum has moved. Alternatively, the best point in the simplex could be replaced instead of the worst, using Equation (2.3). Figure 2.5 illustrates the simplex being contracted by replacing the best point. If a disturbance to the screen happened anytime time after the first three points of the simplex were set and evaluated, the simplex would continue to think that point (1) the best point of the first three, is in the region of the optimum. The performance of point (1) is never retested and its value as a point in the simplex (1-2-3) is assumed to be the same as a point in the simplex (1-5-6).

It is possible the value of point (1) does remain constant, but in an industrial setting, it is more likely that the operating conditions will change. In the event that the optimum location hasn't changed, then this procedure isn't as efficient as the previous method, but does guarantee that a change in optimum location is recognized by the NOFS method, and the optimum is located as time goes on.

$$P_{new,i} = 0.25 * (\sum_{j=1}^{n+1} P_{j,i} + P_{ilo,i}) \quad (for\ i = 1\ to\ n) \quad (2.3)$$

Figure 2.4 shows how the simplex is reduced in size from simplex (1-5-6), to the reduced simplex (1-6-7). After the simplex has been reduced, the simplex cannot reduce in size for at least  $M$  iterations. This procedure ensures that the simplex doesn't shrink in consecutive iterations.

If the simplex is not to be expanded, shrunk, or the second worst point is to be replaced, then the simplex is simply reflected away from the worst point. The operation of reflecting the NOFS is unchanged and the new point is calculated using Equation (1.41). Reflection of the simplex is the same for either the equilateral triangle or the elongated triangle.

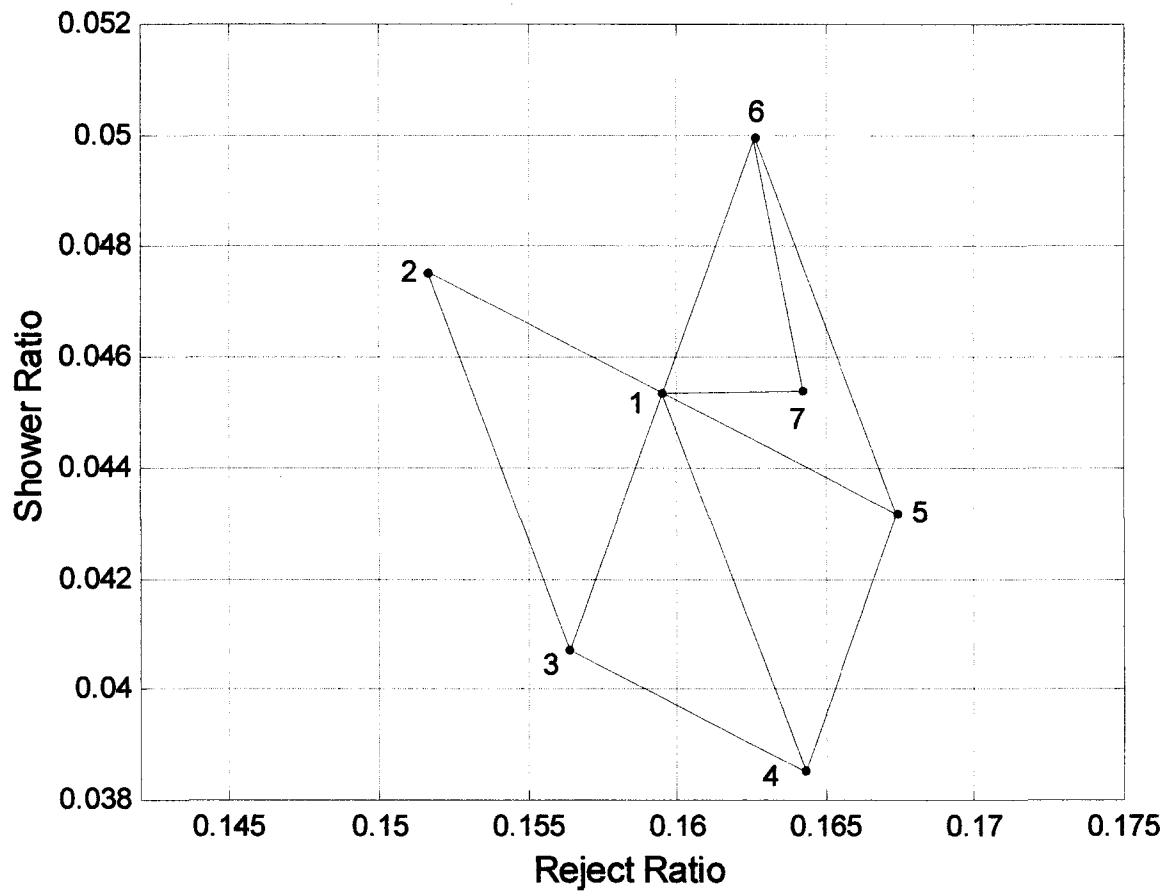


Figure 2.4. Contraction of the simplex by replacing the worst point using the NOFS method.

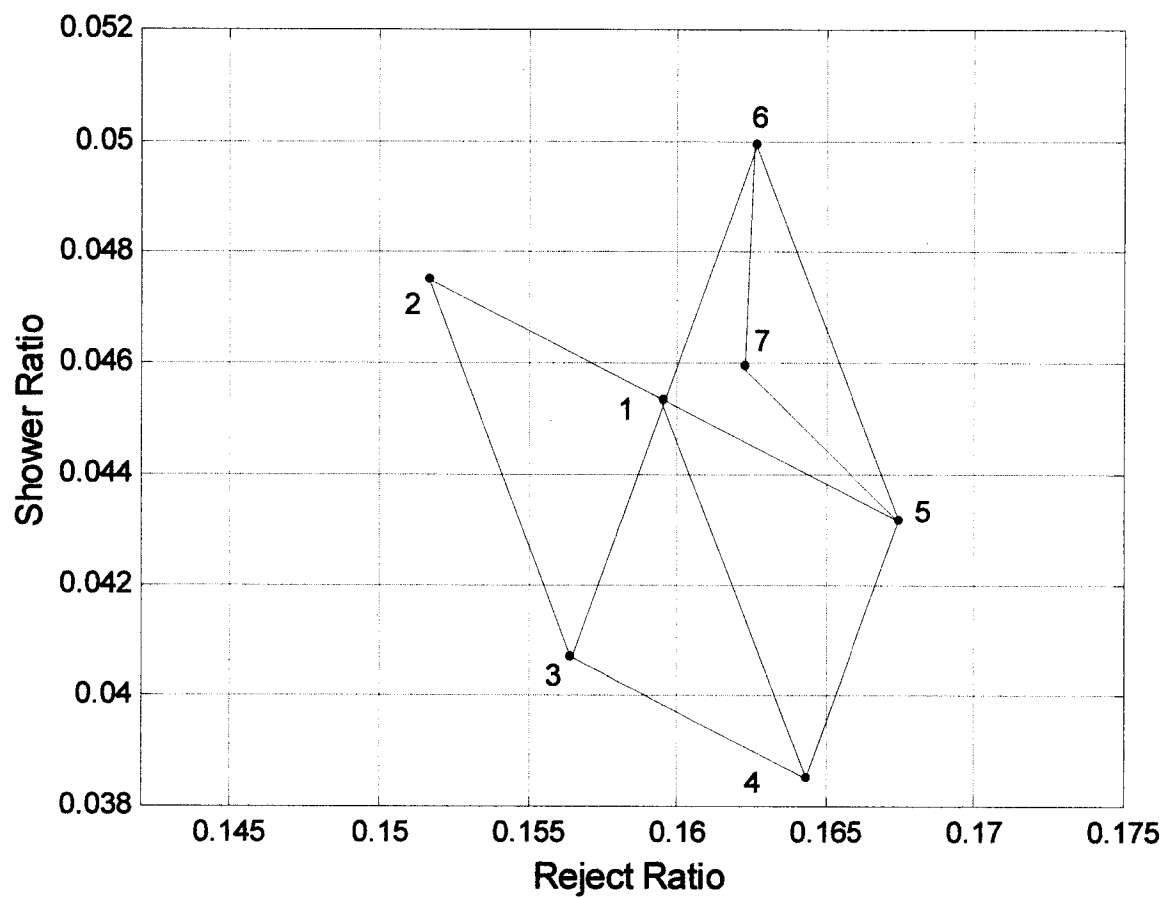


Figure 2.5. Contraction of the simplex by replacing the best point using the NOFS method.

## 2.2 Optimization of a Known Equation

In order to demonstrate that the NOFS method is capable of searching out the optimum operating conditions for the refined rejects pressure screen, it should be tested on a multivariable system where the optimum location is known. Using the algorithm for the NOFS method described in the previous section, the simplex was tested using Equation (2.4). The C code used to simulate the optimization is listed in Appendix A. The code used to simulate the simplex algorithm was adapted from the amoeba function.[18]

$$Perf = 500 * (x_1 - \bar{x}_1)^2 + 500 * (x_2 - \bar{x}_2)^2 \quad (2.4)$$

where  $\bar{x}_1$  and  $\bar{x}_2$  are the optimum values.

The most basic multivariable optimization is to search for a static optimum. Figure 2.6 illustrates the simplex as it searches for the optimum set of conditions. The simulation shown in Figure 2.6 was created using the file simplex1.c, which can be found in Appendix A. The simplex is able to find and begin to pivot the optimum in under 18 samples. Figure 2.6 illustrates how the simplex can track the optimum from a starting location far from the optimum. Some of the simplex variables are as follows:

- starting side length = 0.0035
- minimum side length = 0.0001
- maximum side length = 0.007

Figure 2.6 shows how the simplex can track a static optimum, but in an industrial setting, the optimum set of operating conditions can change on a regular basis. To simulate a moving optimum, the values for the optimum were altered. The optimum values of  $x_1$  and  $x_2$  are entered into the simulation as inputs, and the changes to the optimum were simulated as disturbances by selecting the time and size of the disturbance. The optimum location for  $x_1$  was decreased from 0.14 to 0.08 at the 50<sup>th</sup> sample, and back to 0.14 at the 100<sup>th</sup> sample. The optimum value of  $x_2$  was decreased from 0.07 to 0.04 at the 50<sup>th</sup> sample, and increased to 0.07 at the 100<sup>th</sup> sample, as shown in Figure 2.7.

The results shown in Figure 2.7 demonstrates the ability of the simplex to track a moving optimum. Although the simplex does track the moving optimum, the response to the changes is slow, needing at least 50 samples to reach the new optimum location.

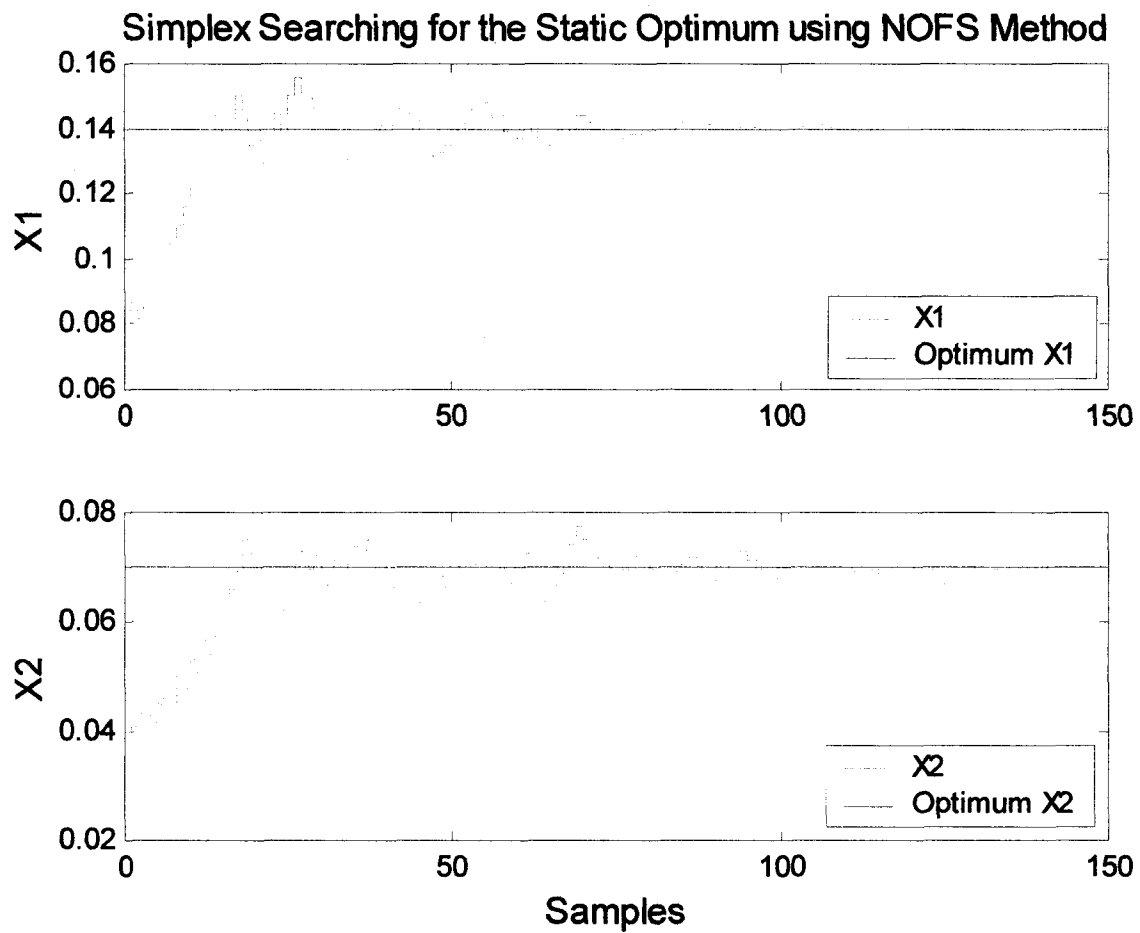


Figure 2.6. Use of the NOFS method to search out a known optimum.

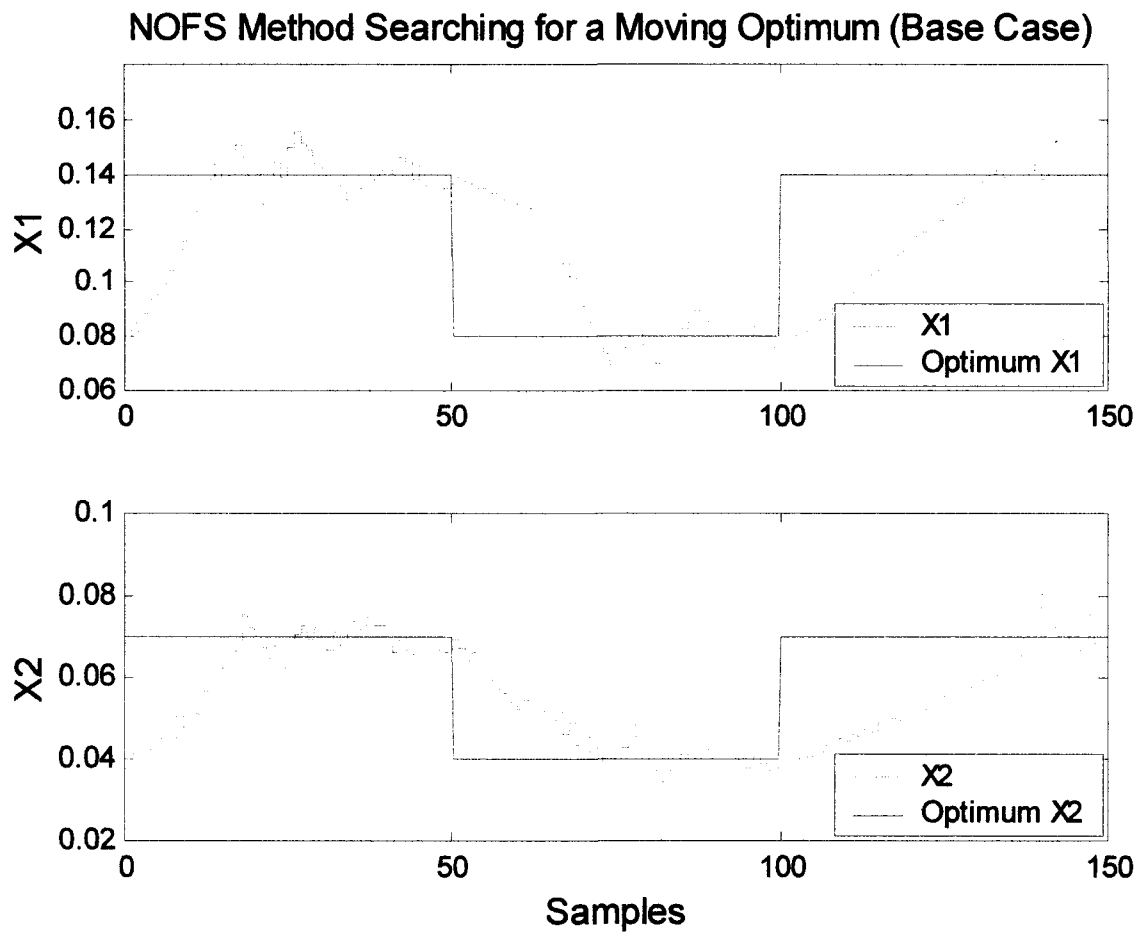


Figure 2.7. Searching for the moving optimum.

One method to improve the speed of the response is to increase the size of the starting simplex. The larger the simplex, the greater the area that is covered with each reflection. Figure 2.8 shows the trend plot of the variables with the starting side length increased from 0.0035 to 0.006. The three main operating variables of the simplex used to create the plots shown in Figure 2.8 are as follows:

- starting side length = 0.006
- minimum side length = 0.0001
- maximum side length = 0.007

The advantage of the larger starting simplex is how rapidly the first optimum is found as compared with Figure 2.7. The disadvantage to the larger starting simplex is the increased overshoot of the first optimum. Since the simplex can expand or contract at anytime during its operation, the effect the starting simplex size is only applicable to the early stages of the optimization.

Beyond the 50<sup>th</sup> sample in Figure 2.8 you would expect that the simplex should look similar to Figure 2.7. The difference between the two figures is that the simplexes are not the same size at the 50<sup>th</sup> sample. Since the sizes are different at the 50<sup>th</sup> sample, all future simplexes will be a different size because each is increased or decreased at the same factor.

Selection of the minimum simplex size can effect the variance of the search for the optimum location. As the simplex size gets smaller, the closer the simplex can get to an optimum. Once the simplex has shrunk to its minimum size, the simplex can only repeatedly pivot its best point until some point when the optimum moves. Figure 2.9 illustrates the results of a simulation where the effect of a larger minimum simplex size was tested. The operating variables used in the simulation are as follows:

- starting side length = 0.0035
- minimum side length = 0.003
- maximum side length = 0.007

With the minimum simplex size increased from 0.0001 to 0.003, the simplex is unable to shrink down to the size of the simplex shown in Figure 2.8. This leads to larger oscillations about the optimum, but does allow the simplex to respond much quicker to a change in optimum. There is a trade-off between fast response and large oscillations that must be considered when selecting the operating variables for the

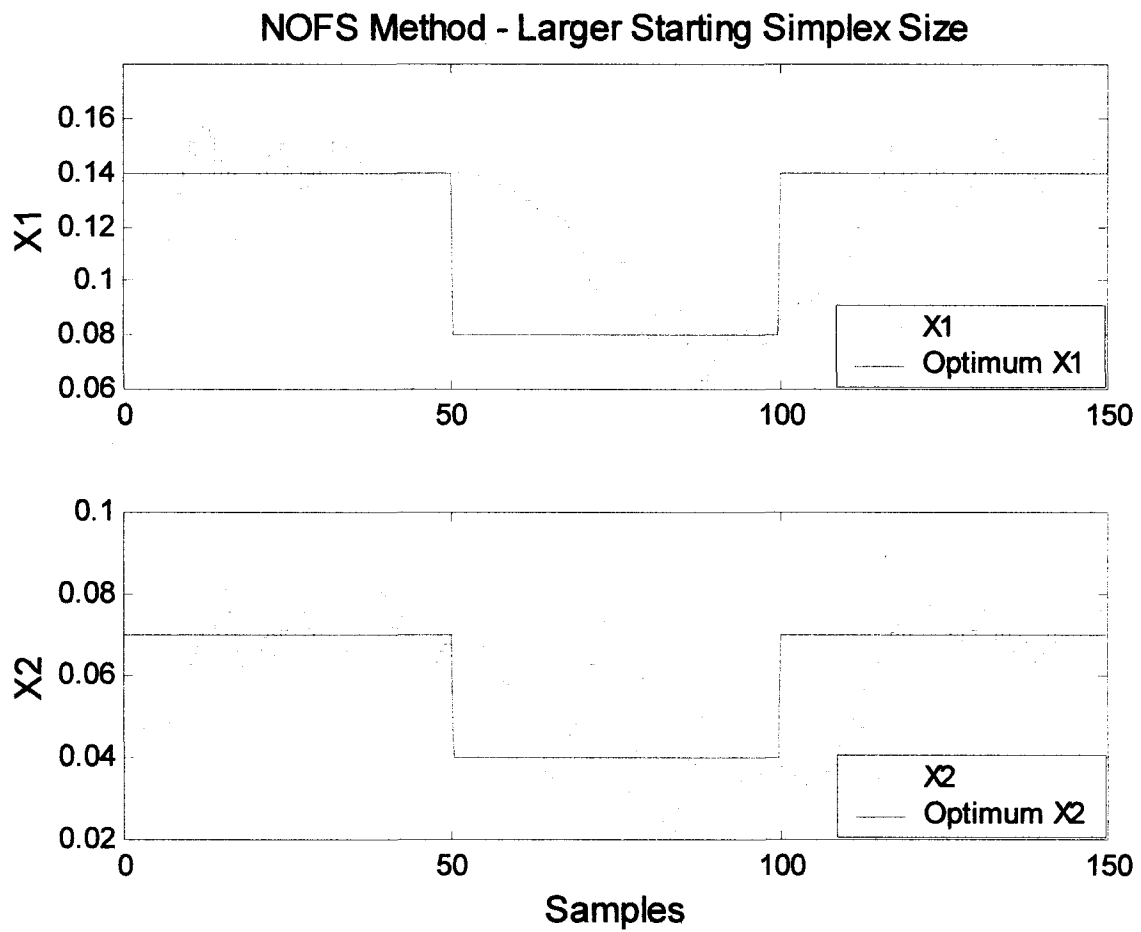


Figure 2.8. The effect of a larger starting simplex size when searching for the optimum.



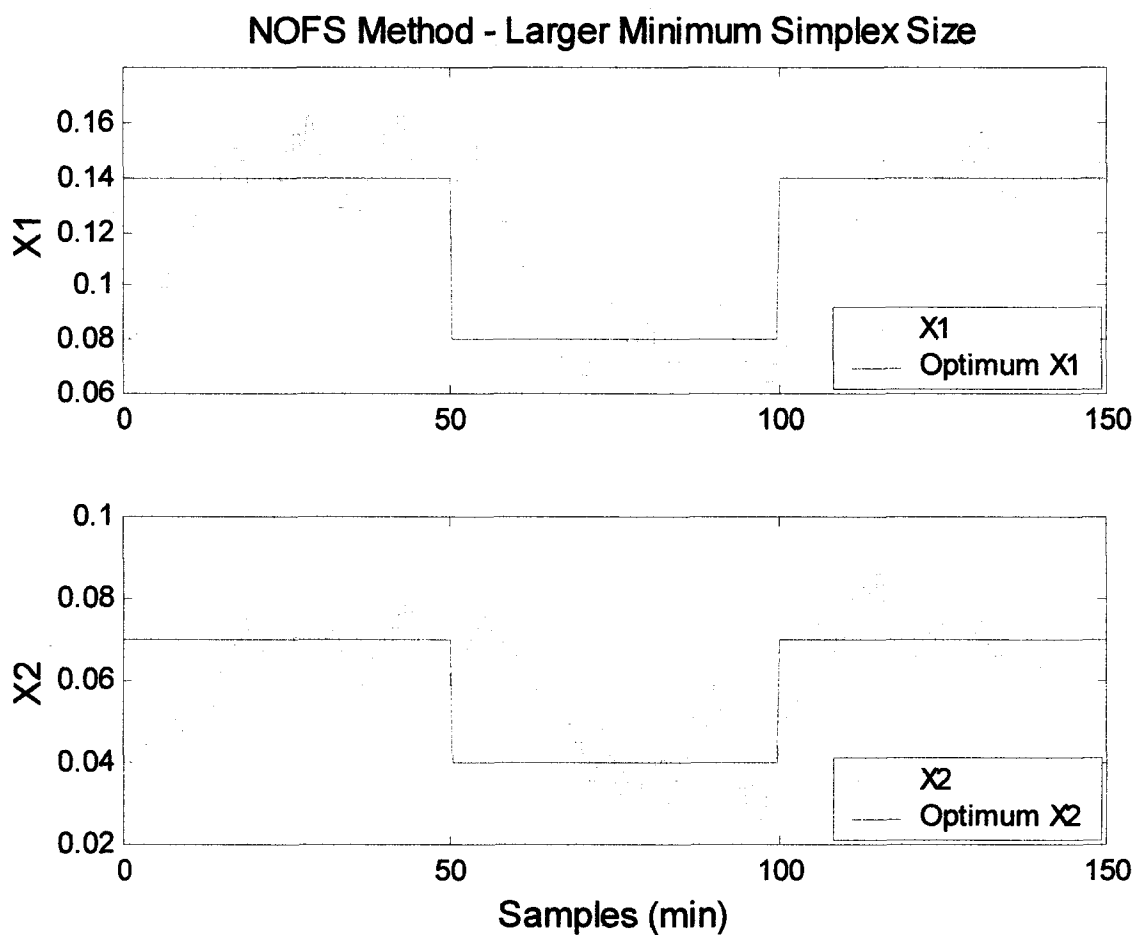


Figure 2.9. The effect of a larger minimum simplex size when searching for the optimum.

simplex. The quicker response to a change in location of the optimum is shown in Figure 2.9. After the 50<sup>th</sup> and 100<sup>th</sup> sample points, the response is quicker in Figure 2.9 than the response in Figure 2.7.

Increasing the maximum simplex size allows for a faster response to a change in optimum. When the optimum moves far from its current location, the simplex with a larger maximum is able to expand several times in order to locate the new optimum. Table 2.4 lists the operating variables for the simulation illustrated in Figure 2.10.

- starting side length = 0.0035
- minimum side length = 0.003
- maximum side length = 0.04

The simulation shown in Figure 2.10 is very similar to simulation illustrated in Figure 2.9. These two plots are similar because of the shape of the simplex during times when expansion would be desired. The simplex is supposed to expand if the newest point becomes the best point. However, the simplex is not permitted to expand if the distance from the centroid to the best point is more than 1.7 times the distance from the centroid to either of the other vertices. Therefore, this rule doesn't allow the simplex to become too elongated in one direction.

It is possible that a larger maximum simplex size can have a slower response. If the simplex is oscillating about the optimum, it is possible for the simplex and the optimum to simultaneously move in opposite directions. If this scenario occurs, the response of the simplex will take longer, as long as the optimum only moved a small distance.

A small section of the simulation shown in Figure 2.10 is split into each step are plotted in Figures 2.11(a) through (g). During the time period from 50 to 100 samples., the location of the optimum was (0.08, 0.04), and is shown by an X. The first simplex shown in (a) is a product of several other simplex movements, and as a result, the second worst point is replaced, creating the second simplex shown in (a). Since the newest point in the simplex is neither the best or the worst, the movement is to replace the worst point, creating the third simplex shown in (a). The simplex is moving directly towards the optimum, resulting in the newest point becoming the best. Therefore, the simplex should expand, except that the shape prevents the simplex from elongating any further. This causes the simplex to reflect away from the worst point, creating the fourth simplex shown in (a). The fourth simplex faces the same situation, so it

reflects to create the first simplex shown in (b). The newest point is further from the optimum than the newest point in the fourth simplex in (a), causing the first simplex in (b) to reflect away from the worst, creating the second simplex illustrated in (b). The following movement is away from the optimum, but the simplex cannot go backwards, therefore the second worst point is replaced, creating the third simplex shown in (b). At this time, the cost of the point closest to the optimum has been the best point in the last four simplex's. As a result of this, the simplex should contract in an attempt to close in on the optimum. Unfortunately the current size of the simplex is smaller than the specified minimum, so the simplex cannot contract, therefore it must reflect away from the best point, creating the fourth simplex shown in (b). When the best point is replaced, the simplex is moving away from the optimum. Moving away from the optimum is a short term concern, but the long term progression towards the optimum is continued. If the worst point of the simplex were replaced, the simplex would continue to pivot about the point closest to the optimum without ever detecting a shift in the location of the optimum. This method of replacing the best point causes the oscillations about the optimum to be larger, but guarantees that the simplex will recognize any changes in the location of the optimum.

From the fourth simplex in (b), the second worst point is replaced, creating the first simplex in (c). At the same time, the location of the optimum has shifted to (0.14, 0.07), resulting in an increase in cost for the newest point. Since the newest point becomes the worst point, the second worst point is replaced in the simplex. Replacing the second worst point and reflecting creates the second simplex shown in (c). With the creation of the second simplex in (c), the scenario of four consecutive movements having the same best point is encountered again. Since the size of the simplex hasn't changed, the simplex still cannot contract. Therefore the best point is replaced, and the third simplex shown in (c) is created. At this point the simplex is searching for the location of the new optimum, and the simplex is reflected to produce the fourth simplex shown in (c). The newest point in this and the following simplex are each the best points causing the simplex to want to expand. Due to the shape of both simplexes, neither simplex is permitted to expand. This allows each of them to reflect by replacing the worst point, creating the second simplex shown in (d). The newest point in the second and third simplexes of (d) aren't the best, therefore each is reflected away from the worst point, creating the fourth simplex shown in (d).

Due to the change in alignment with respect to the location of the optimum, the third simplex in (d) is expanded to create the first simplex (e). Since the newest point is the best point, and the shape meets the requirements, the simplex is expanded to produce the second simplex shown in (e).

The movements of the simplex shown in Figure 2.11(a) through (g) illustrate an example of how the simplex attempts to close in on an optimum, along with its ability to recognize a change in the location of the optimum.

Changing the size of the ratio of distances from the centroid to the vertices alters the criteria for expansion or contraction. For expansion, increasing the ratio of the distance from centroid to best point, to the distance from centroid to any other point, allows for the simplex to become more elongated if the expansion is in the direction of the point furthest from the centroid. In all of the previous simulations the ratio of distances was 1.7. The ratio of distances for the simulations shown in Figures 2.12 and 2.13 is 2.0. The other operating variables for the simplex are the same as those used to create the simulations shown in Figure 2.6 and 2.7, and are listed as follows:

- starting side length = 0.0035
- minimum side length = 0.0001
- maximum side length = 0.007

The movements of the simplex of Figure 2.12 are illustrated in Figure 2.12. Looking at the movements of the simplex illustrated in Figure 2.13 that there is a problem with the algorithm of the simplex. The path laid out by the simplex in search of the optimum, shown by the X, isn't very direct or quick, but does eventually find the optimum location. The reason for the indirect pathway is the restrictions placed on the size and shape of the simplex. The simplex is created from point (1) (0.08,0.04) by calculating points (2) and (3). In the process of calculating the second and third points of the starting simplex, the initial simplex is reflected away from the worst of the three starting points to get the optimization started. Inspection of the simplex (2-3-4) would under normal circumstances cause the simplex to expand beyond point (4), but because the simplex is just starting, it cannot expand due to a rule that prevents the simplex from expanding immediately following a contraction. This criteria does prevent the expansion during the early stages of the optimization, but doesn't interfere at any other time. From the simplex (2-4-6) through to (11-12-13) the simplex is trying to expand. The expansion isn't allowed because

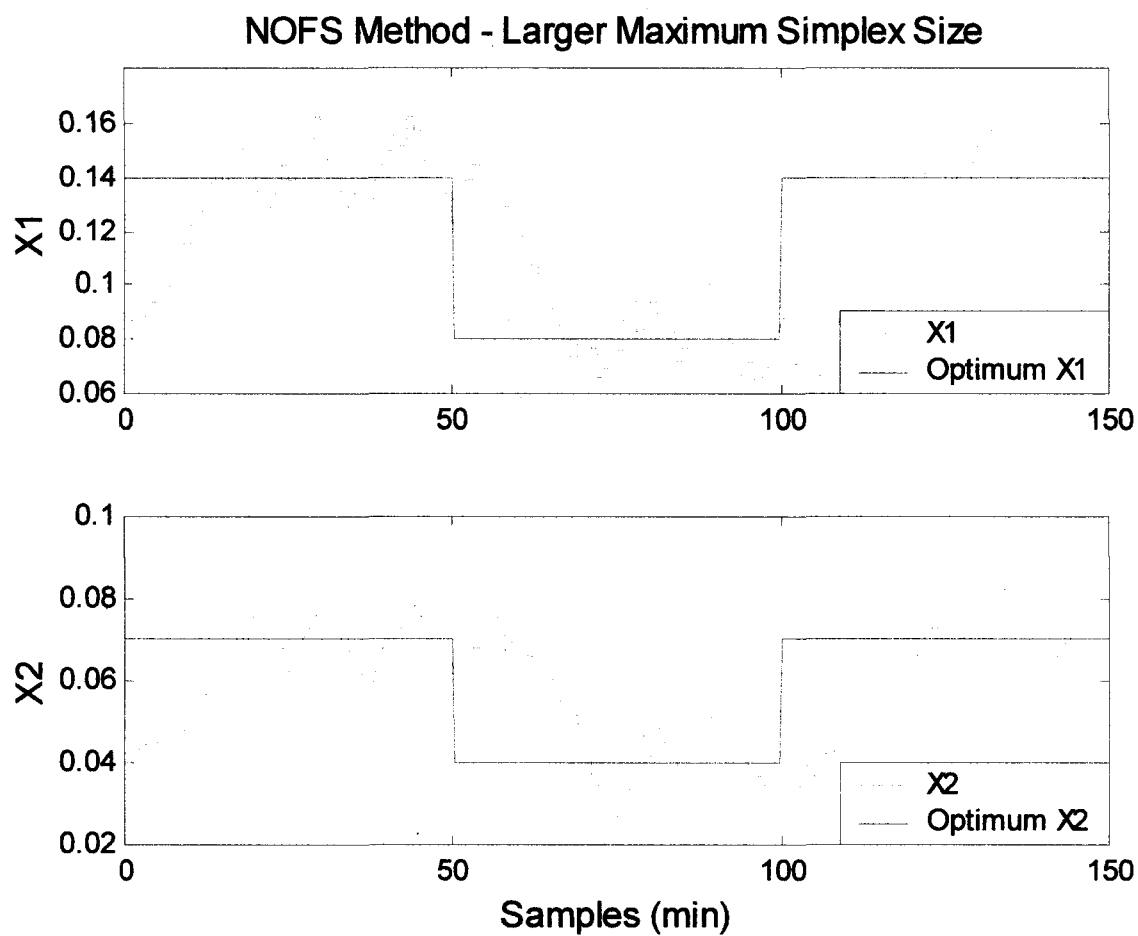


Figure 2.10. The effect of a larger maximum simplex size when searching for the optimum.

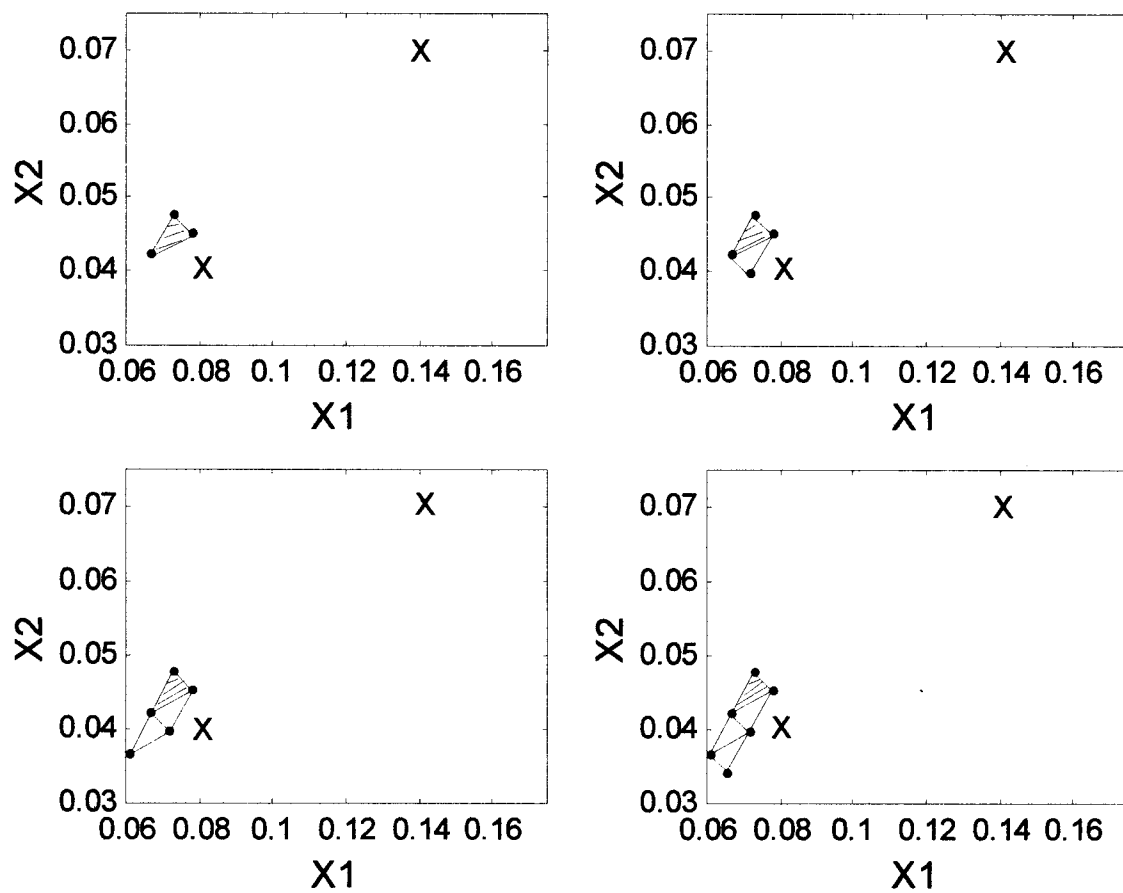


Figure 2.11(a). Tracking the progress of the simplex in Figure 2.10.

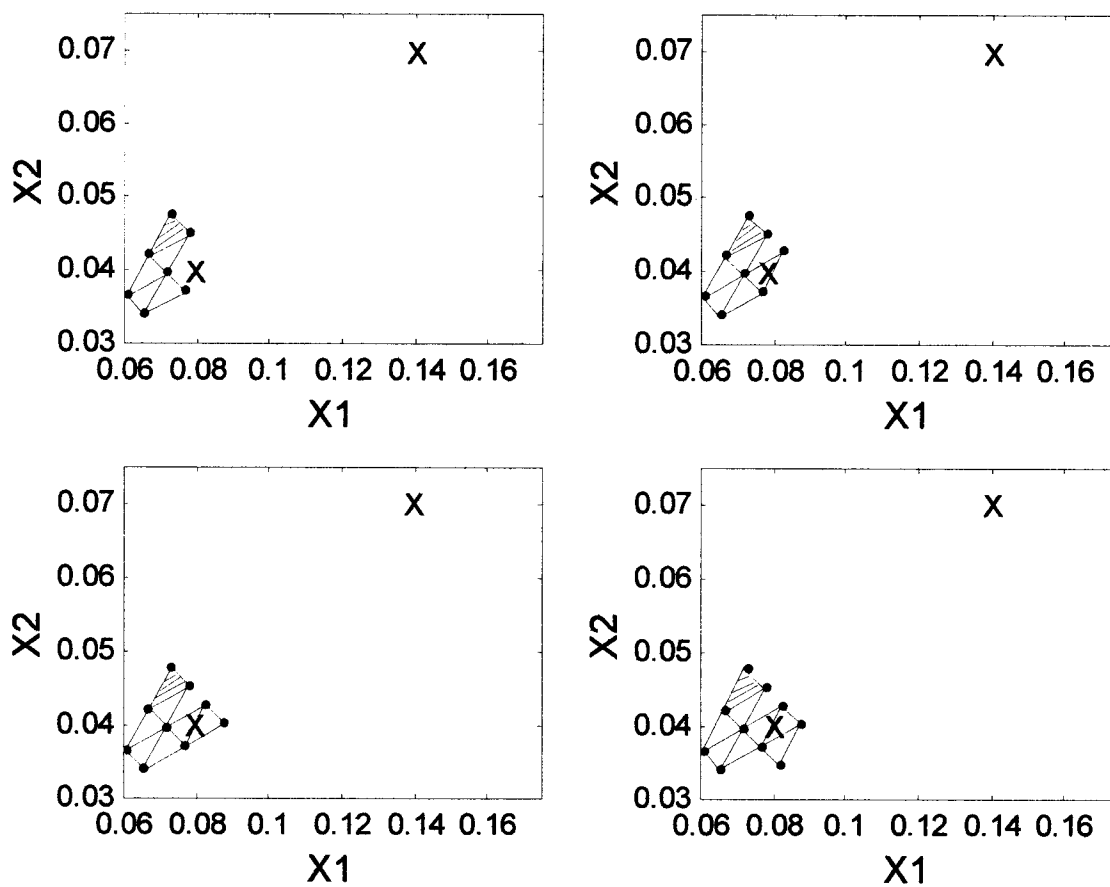


Figure 2.11(b). Tracking the progress of the simplex in Figure 2.10.

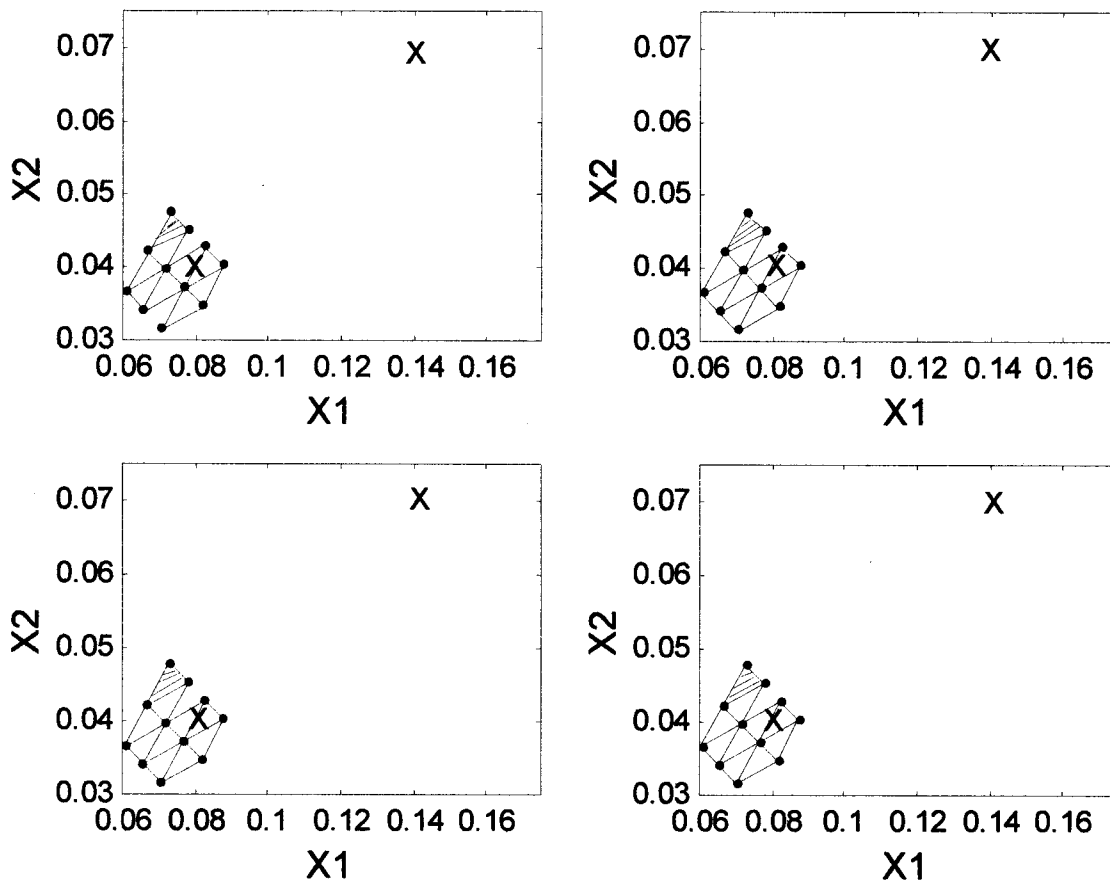


Figure 2.11(c). Tracking the progress of the simplex in Figure 2.10.



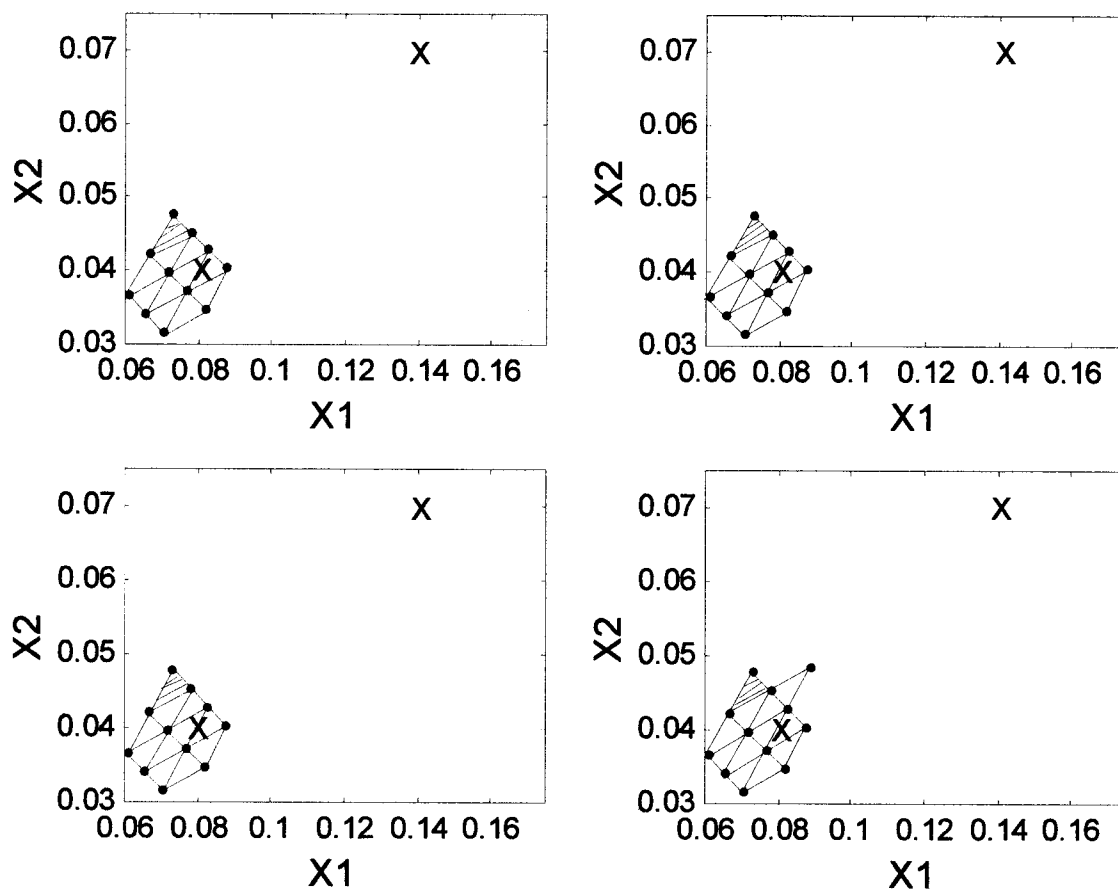


Figure 2.11(d). Tracking the progress of the simplex in Figure 2.10.

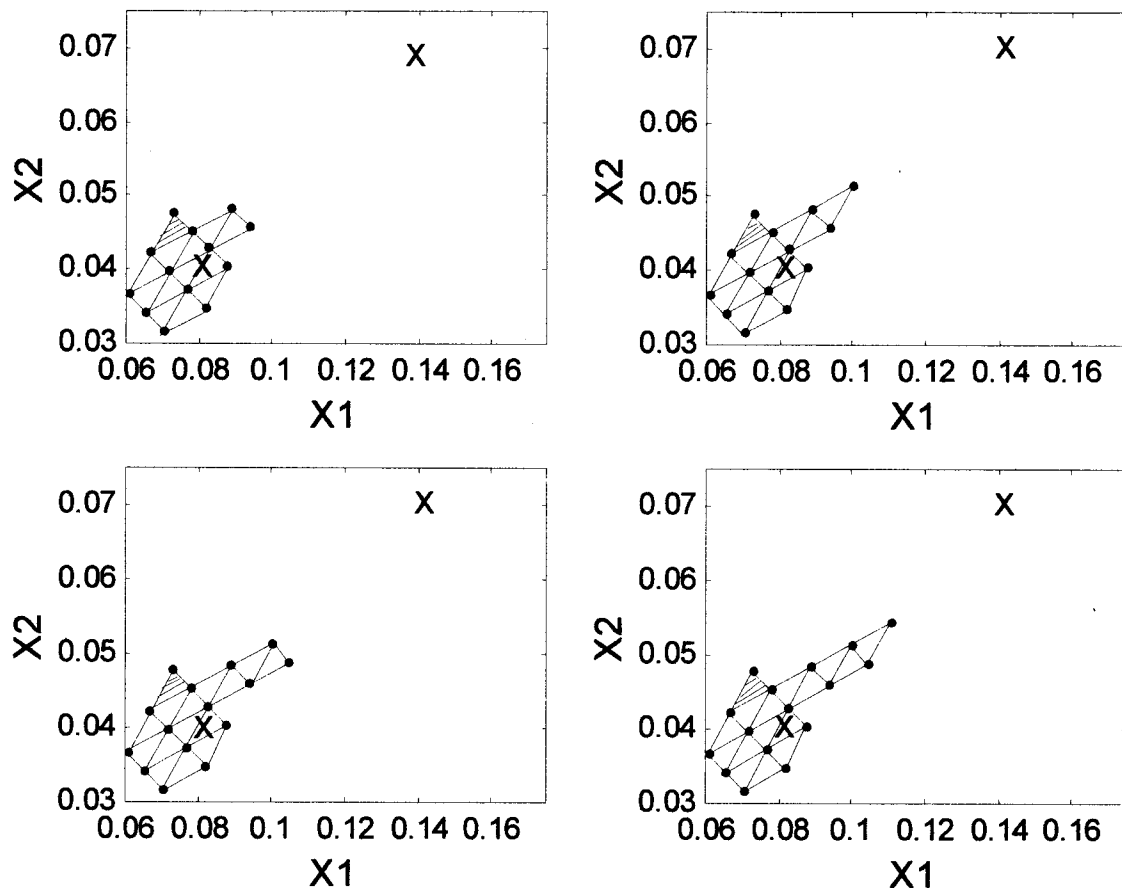


Figure 2.11(e). Tracking the progress of the simplex in Figure 2.10.

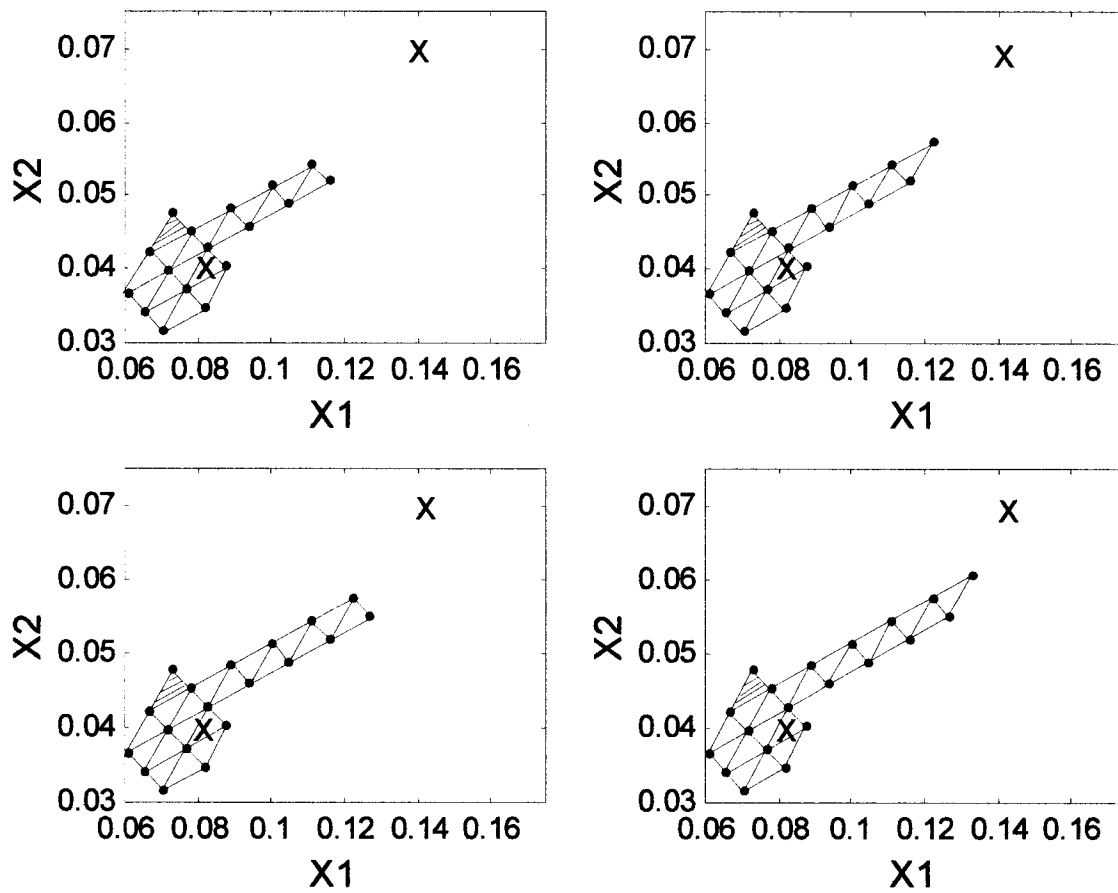


Figure 2.11(f). Tracking the progress of the simplex in Figure 2.10.

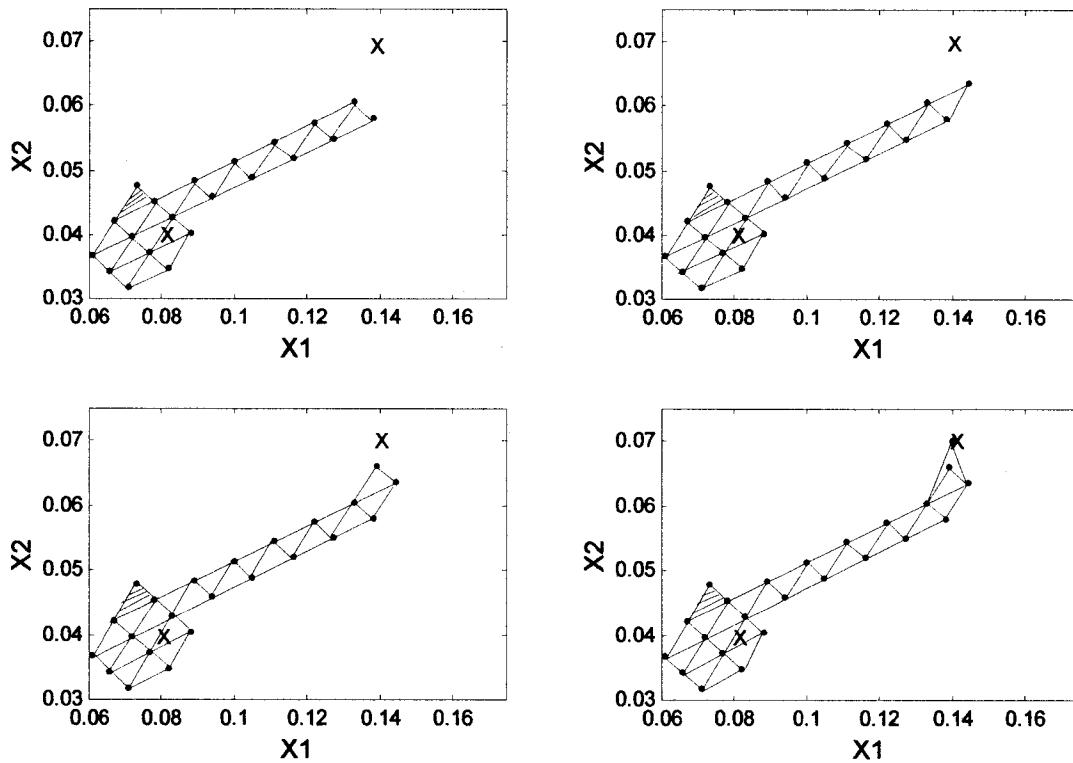


Figure 2.11(g). Tracking the progress of the simplex in Figure 2.10.

of the size of the simplex. Since the maximum size is 0.007 and the actual maximum length of the simplex is 0.00788, the simplex cannot expand. The simplex is also distorted by the scaling of the axis. The length of the X1 axis is much longer than the X2 axis, creating a compacted simplex in the X1 direction.

From the simplex (11-12-13) to (13-14-15), the point (13) is the best point. Since the best point in the simplex (13-15-16) is point (16), the simplex isn't contracted, but is reflected to create the simplex (13-16-17). The simplex then continues to progress towards the optimum location in the following simplex steps.

Using the larger distance ratio for the simplex simulation illustrated in Figure 2.12 doesn't provide better results than the simulation shown in Figure 2.7. The larger ratio creates larger oscillations because of the elongation of the simplex, without improving the response time. Figure 2.14 illustrates the effect of increasing the maximum simplex size and using a ratio of 2.0. The simplex becomes too large and elongated in the X2 direction. Just prior to the 125<sup>th</sup> sample, the movement of the simplex in both directions is greater than it should be for ideal optimization. Therefore, the operating conditions used in this simulation should be altered for better optimization. Some of the key variables used in the simulation to create Figure 2.14 are the following:

- starting side length = 0.0035
- minimum side length = 0.003
- maximum side length = 0.04

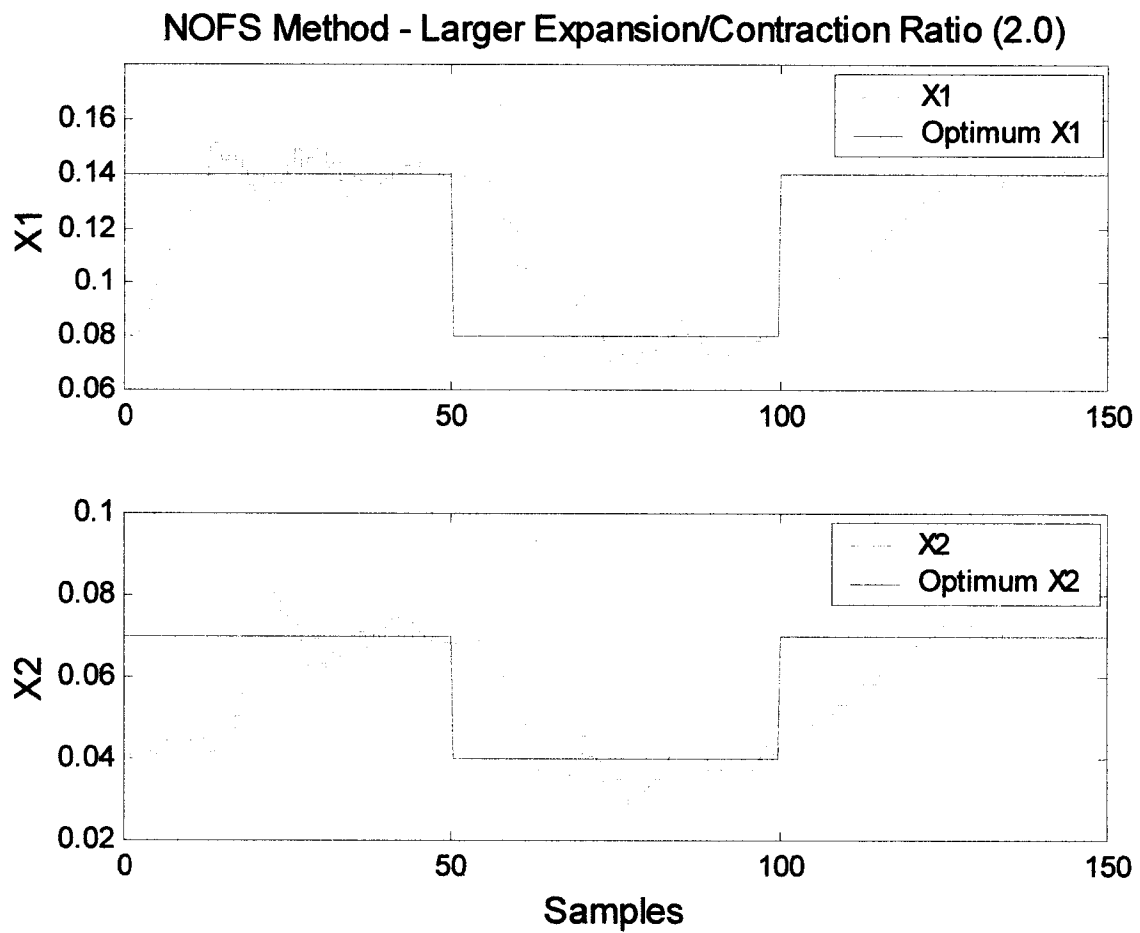


Figure 2.12. The effect of using a larger size ratio for expansion/contraction when searching for the optimum.

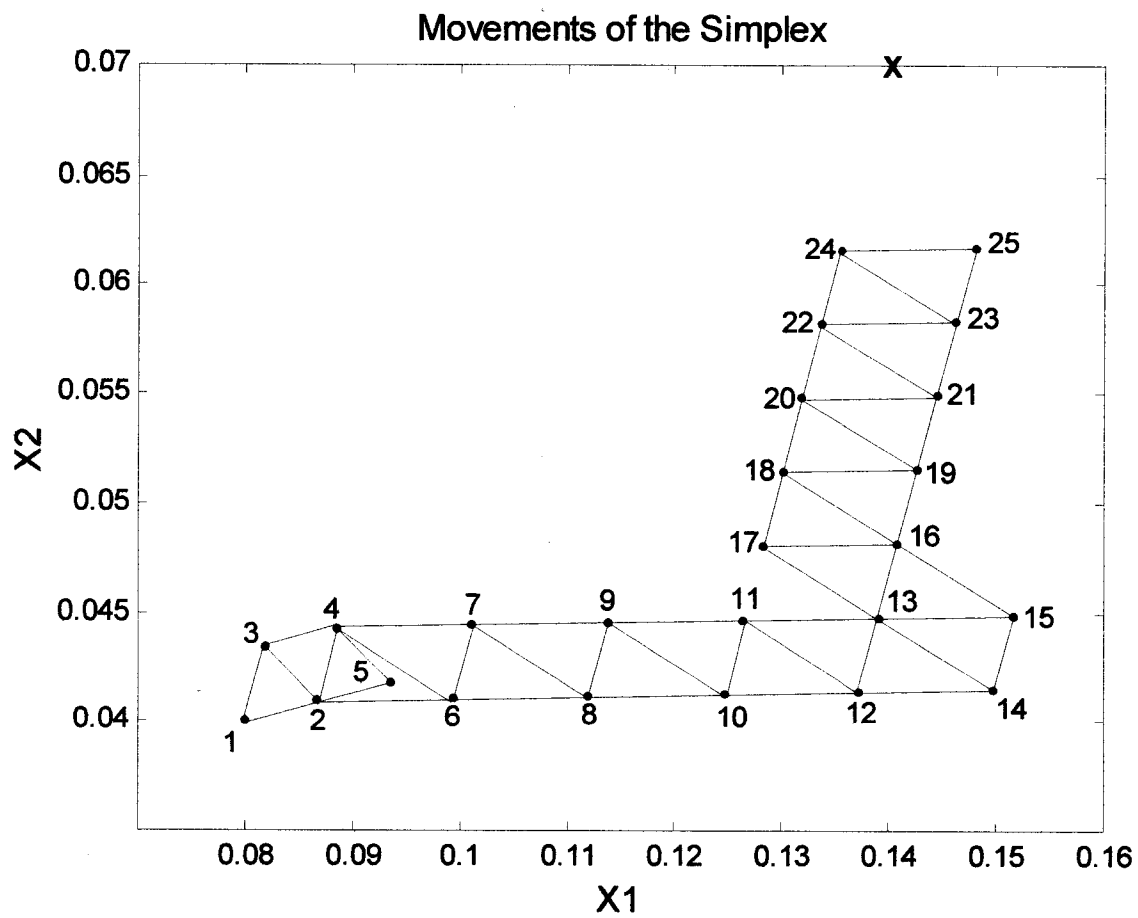


Figure 2.13. Illustrating the movements of the simplex for the simulation shown in Figure 2.12 for the time period 0 to 25 samples.

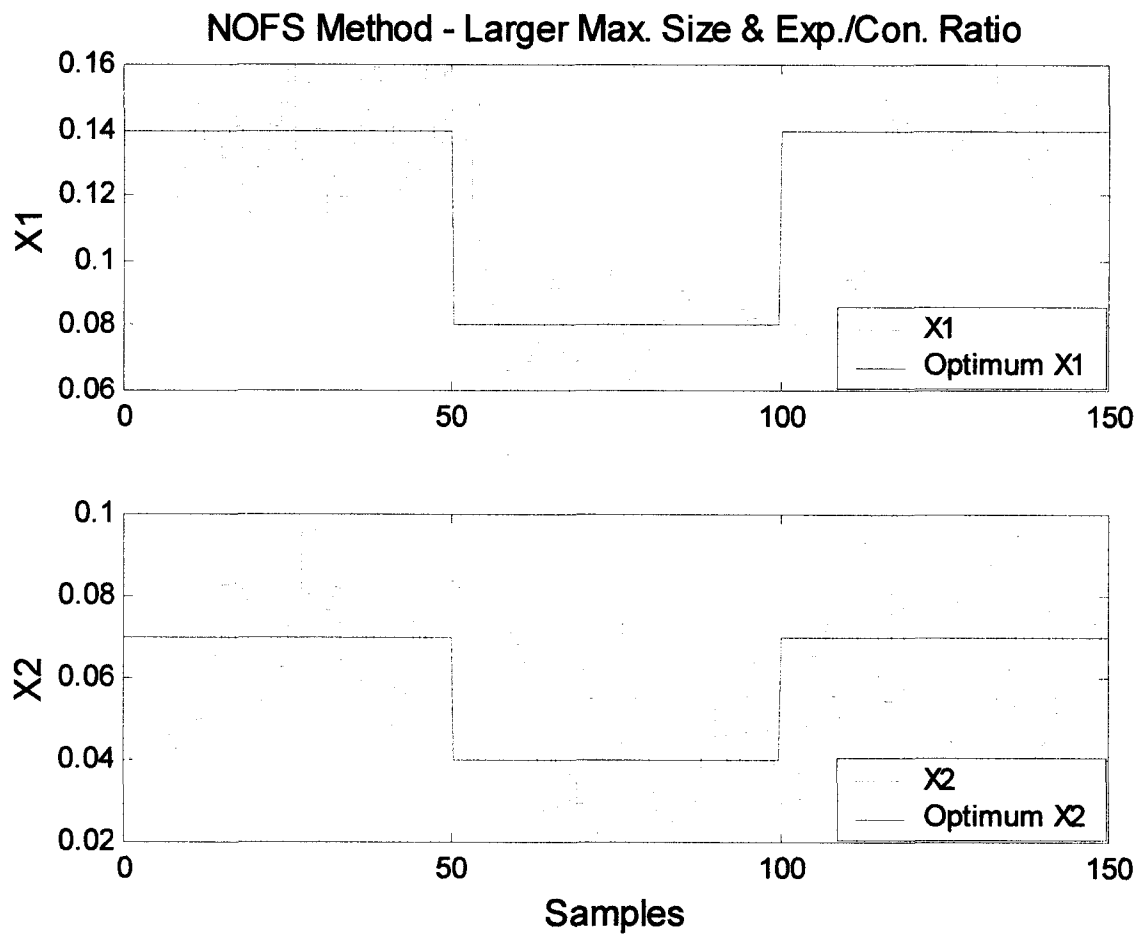


Figure 2.14. The effect of using a larger size ratio for expansion/contraction and a larger maximum simplex size when searching for the optimum.



## 2.3 Optimization using the Equilateral Optimum Following Simplex

The Equilateral Optimum Following Simplex (EOFS) method uses a simplex that remains an equilateral triangle during the process of searching for the optimum. This method can be used as an alternative in searching for the optimum of Equation (2.4). The algorithm for this method is listed in the file `seqsimp.c` in Appendix B. The difference between this method and the previous one is the method for expanding or contracting the simplex. In the previous method the simplex is expanded or contracted by replacing one point in the simplex, thereby altering the size and shape of the simplex. In this method the simplex is expanded or contracted by replacing two of the three points. By replacing two points, the simplex is recalculated in the shape of an equilateral triangle and the size is changed by a selected factor. The method of keeping the simplex equilateral is the method used in the sequential simplex method, except that in order for the SSM to work it had to be modified. The modification is to the contraction step. Similar to the NOFS method, during contraction the best point must be rejected to prevent the simplex from missing any changes to the location of the optimum.

This method is tested using the same data files as the previous method. The first simulation uses the file `seqsimp.c` with the results illustrated in Figure 2.16. The simplex quickly finds the optimum, but doesn't settle down enough. The minimum simplex size is too large for this simulation, and must be decreased if the simplex is to shrink closer to the optimum.

The simulation testing the ability of the EOFS method to locate and track a moving optimum is shown in Figure 2.17. This simulation uses the same conditions as the simulations shown in Figures 2.7 and 2.12. Since this method allows the simplex to grow in two directions at one time, the simplex expands and contracts much quicker than the methods used in the other simulations shown. The response to changes in Figure 2.17 is better than either of the comparable simulations, but this is also a factor of the minimum and maximum simplex size.

Figure 2.18 illustrates the effect of a larger starting simplex on the simplex. This simulation is comparable to the one shown in Figure 2.8. In the first 75 samples the effect of the larger starting size is overshoot and large oscillations for both  $X_1$  and  $X_2$ . Afterwards, the simplex begins to settle down and

respond better to the changes in optimum. The response time is better, but the ability to close in on the optimum is worse than the simulation shown in Figure 2.8.

Figure 2.19 displays the simulation showing the effect of using a larger minimum simplex size. The change in size was negligible which is noticeable by examining both Figures 2.17 and 2.19, they are identical.

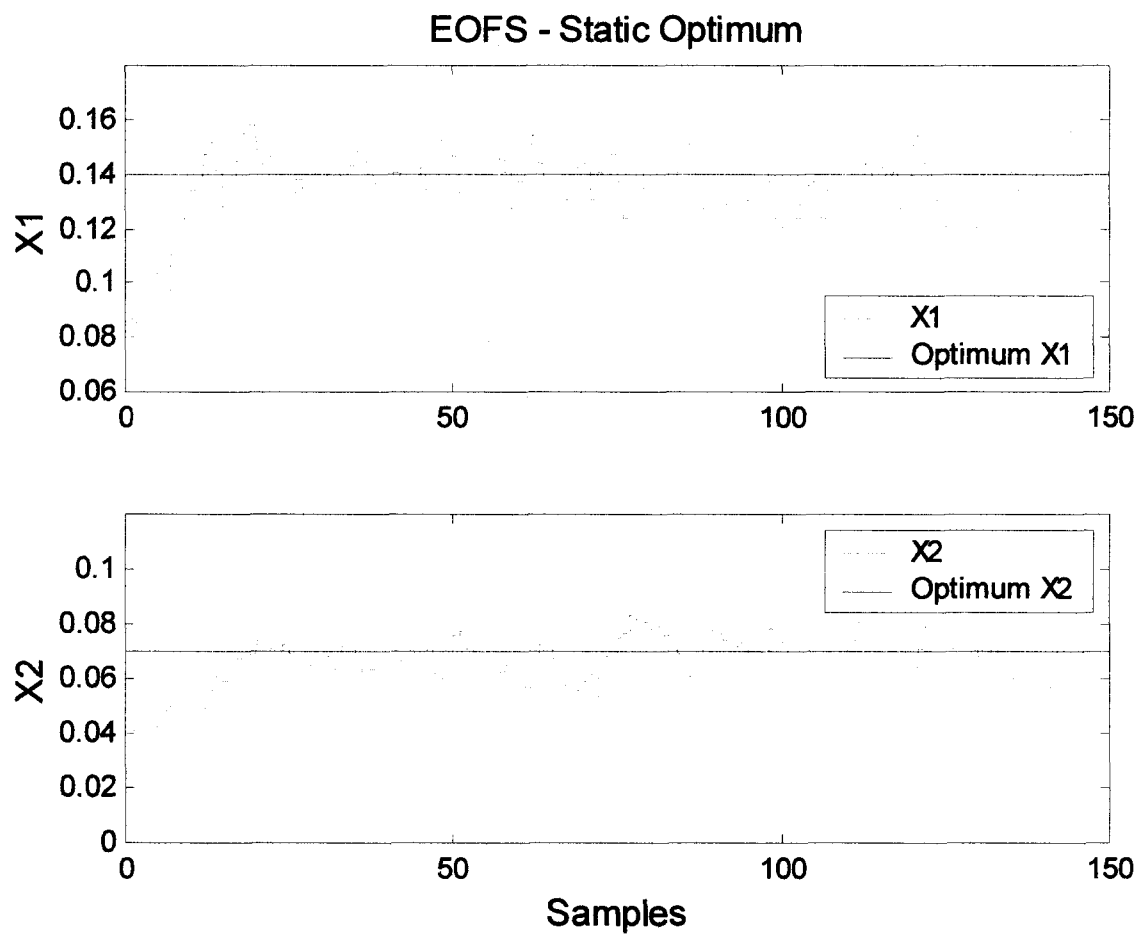


Figure 2.15. Searching for a static optimum using the EOFS method.

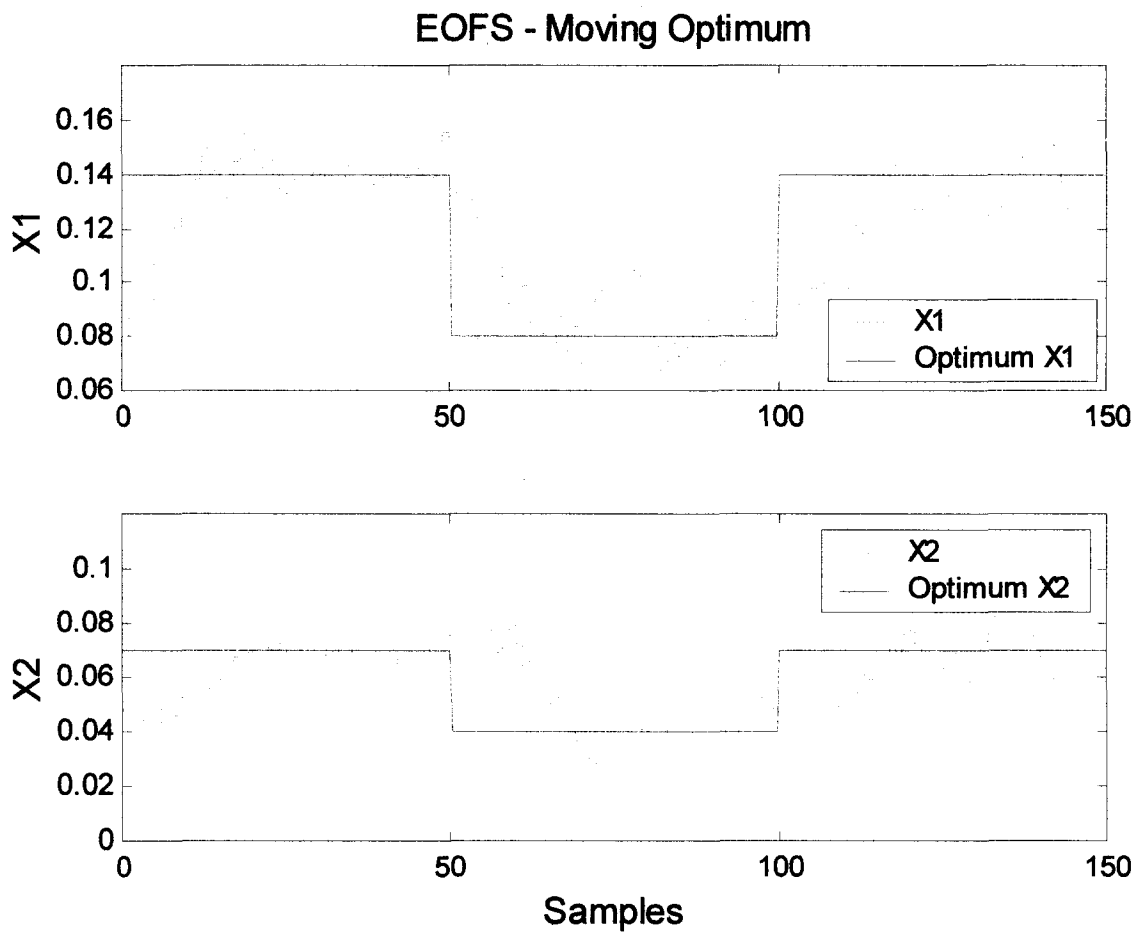


Figure 2.16. Searching for a moving optimum using the EOFS method.

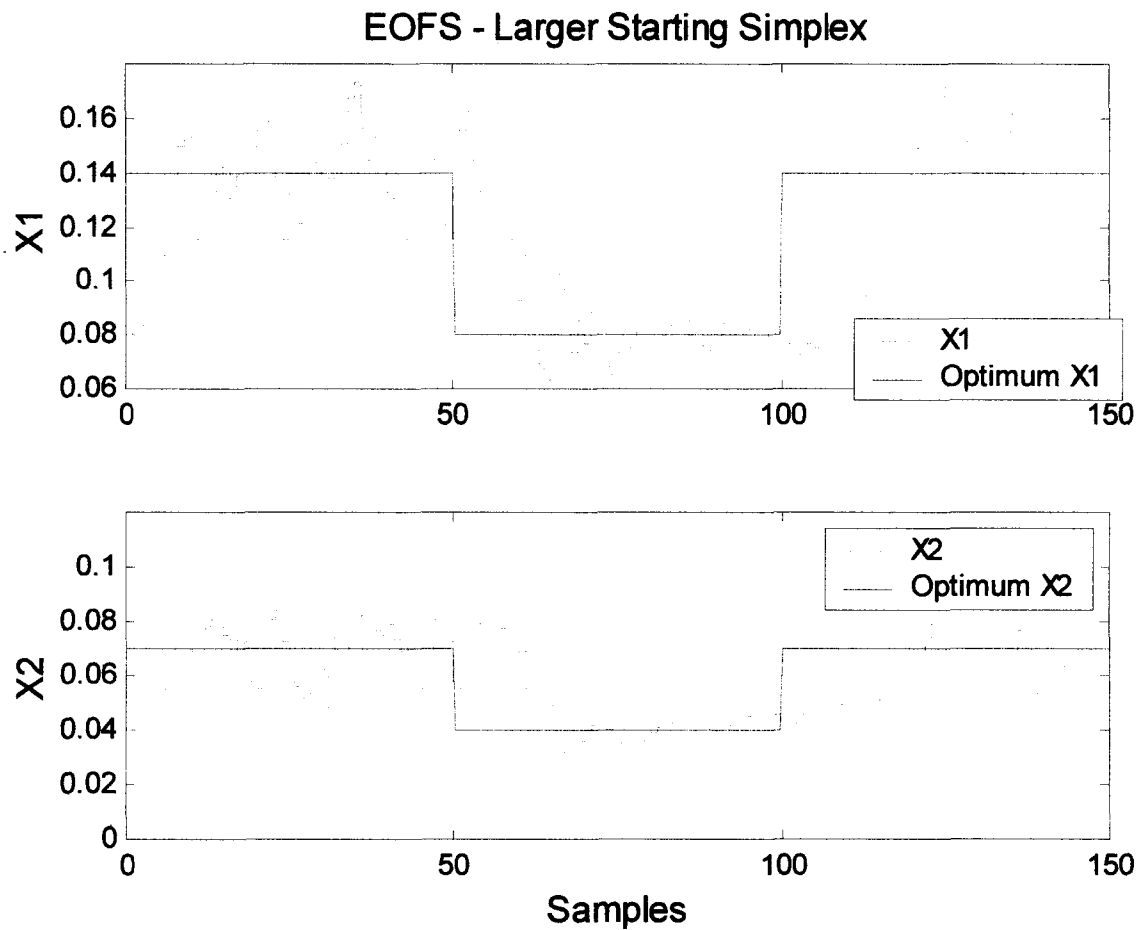


Figure 2.17. Searching for a moving optimum using the EOFS method and a larger starting simplex.

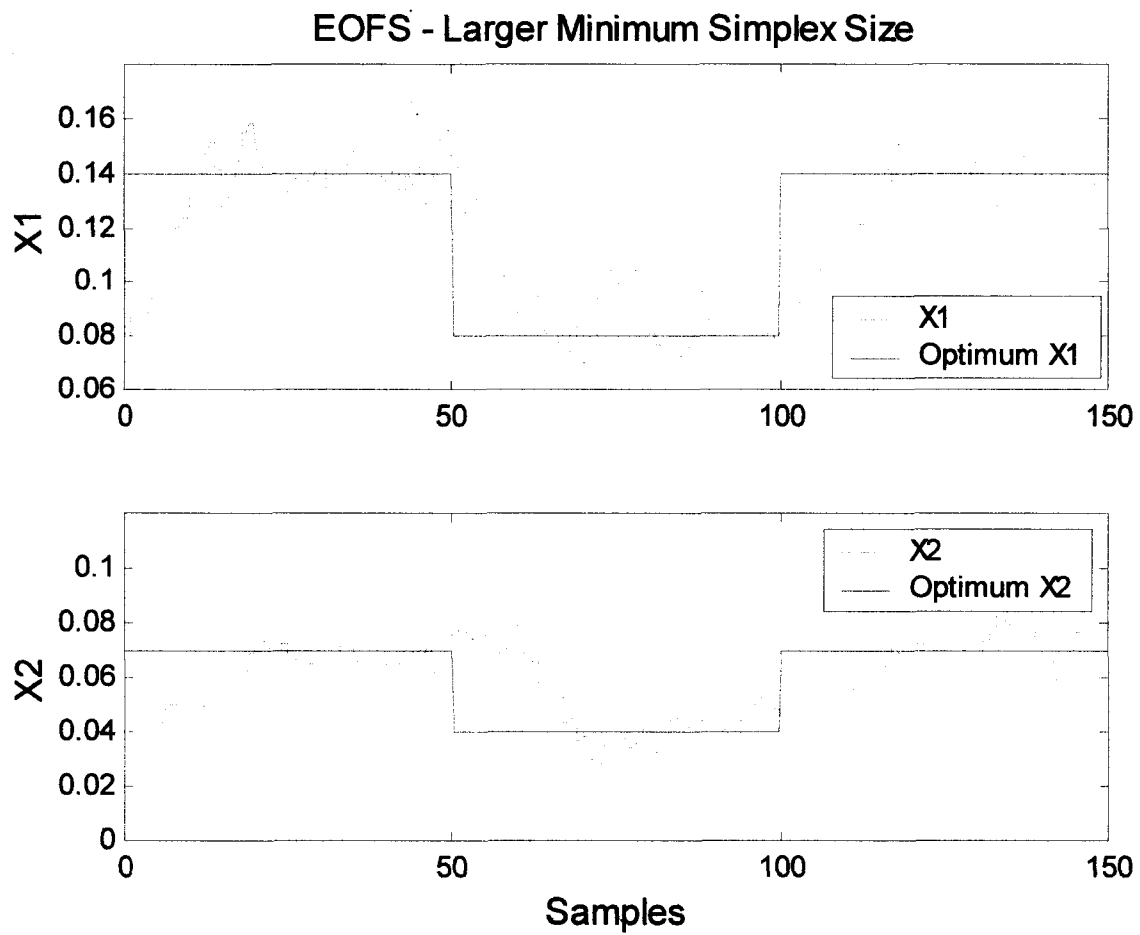


Figure 2.18. Searching for a moving optimum using the EOFS method and a larger minimum simplex size.

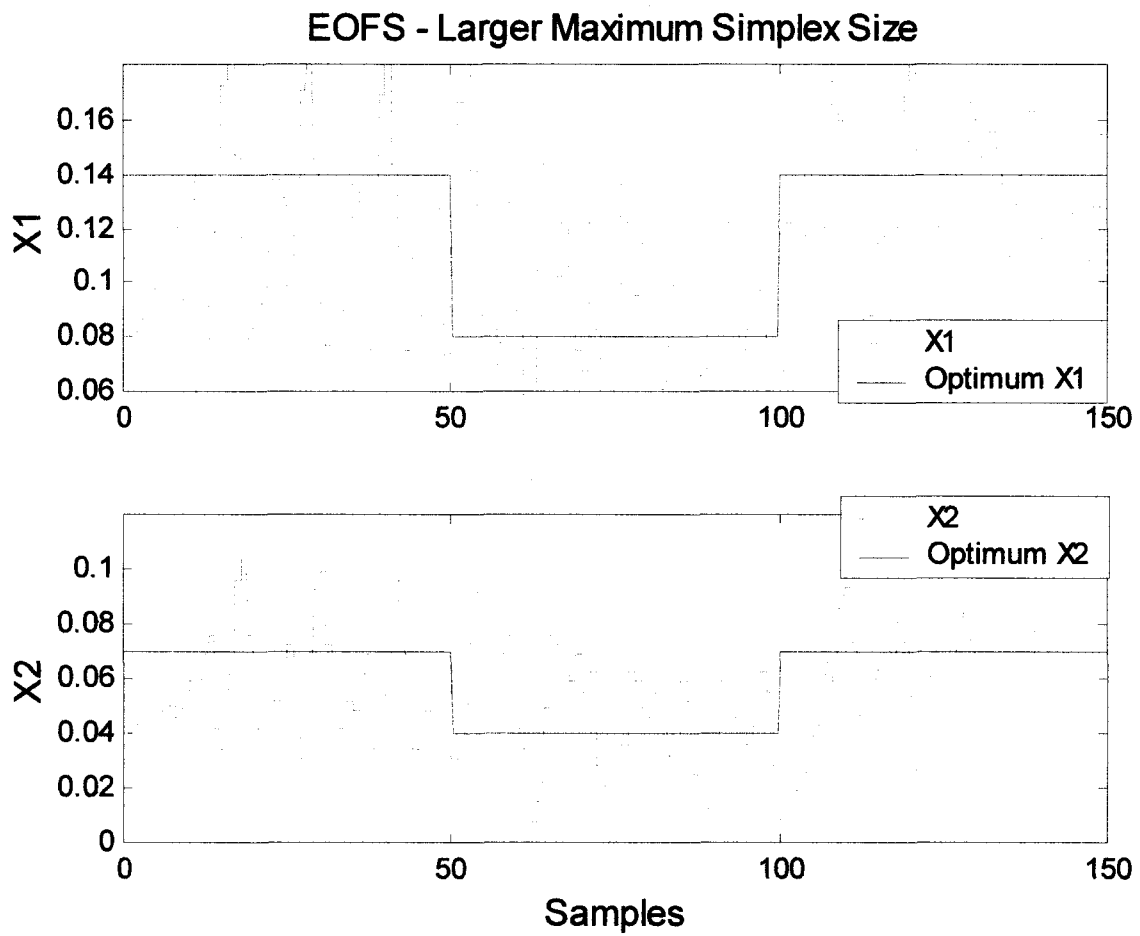


Figure 2.19. Searching for a moving optimum using the EOFS method and a larger maximum simplex size.

The simulation shown in Figure 2.19 illustrates the effect of using a larger maximum size. Unfortunately, the size is too large, and the results are poor. Since the size of the oscillations are so large, the simplex never gets the chance to shrink towards the optimum before the optimum shifts to a new location.

## **2.4 Comparing the Optimum Following Simplex Methods**

The EOFS method is very similar to the sequential simplex method. The difference is the EOFS method has the ability to track a moving optimum. For the simplex to track a moving optimum a step had to be added to ensure the simplex would respond to changes in the location of the optimum. This step was to reject the best point when the simplex was to shrink. Rejecting the best point prevented the simplex from rotating about a single point indefinitely. If the optimum does not move during this period then the rejection of the best point would slow the progress of the simplex, but would ensure the long term success and tracking of any changes to the optimum. Otherwise, the only other difference between the sequential simplex method and the EOFS is the expansion and contraction factors. Two different sets of factors were used during the simulations to determine the best ratio for expanded and contracting the simplex.

The NOFS method is an evolution of the Nelder and Mead method. The advantage of the Nelder and Mead method over the sequential simplex method is its ability to accelerate towards the optimum by expanding in one direction only. Rapid tracking of the optimum is the advantage of the step. The downside to this progression is the irregular shape that the simplex can become. Expanding in one direction only causes the simplex to become elongated in one direction. Once the simplex becomes elongated in one direction the response to a change in optimum can be poor if the optimum moves to a location that is perpendicular to the elongated direction of the simplex. In this scenario the simplex steps would be small unless the simplex continued to expand in the direction of the new optimum. The problem with expanding in the direction of the new optimum is the size of the simplex. Steps need to be taken to prevent the simplex from growing too large. To prevent any of these problems from occurring, three constraints were placed on the simplex movements.

The first constraint was to prevent the simplex from becoming too elongated. Before the simplex is permitted to expand, the shape is checked by measuring the distance from the centroid each of the



vertices of the simplex. If the longest distance is more than 1.7 times longer than the shortest distance, the simplex is not permitted to expand in the direction of the longest line.

The second constraint is on the maximum size of the simplex. If the longest line exceeds the maximum distance, then the simplex is not permitted to expand. The alternative step is to reflect without expanding.

The third constraint is on the minimum size of the simplex. If the simplex is already at the minimum size, the simplex will reflect only.

The NOFS method also includes the step of rejecting the best point when trying to contract the simplex.

Overall the NOFS method provides a new optimization technique that can be useful in tracking a moving optimum. The advantages to using it is that the size and shape can be controlled by the user, and that the response time is fast if the correct range of simplex sizes is chosen. The disadvantage to using the NOFS method is that the simplex can become elongated in one direction, causing the response time to be slow if the optimum is parallel to the elongated direction of the simplex.

The advantage to using the EOFS method is that the shape of the simplex will always remain equilateral, therefore response time is dependent only upon the size and not the shape of the simplex. The disadvantage to this method is the expansion and contraction occurs in two directions, and the simplex can become very large in area in a short time period.

Therefore, the simulations shown in this chapter provide the evidence that both the methods are capable of optimizing a process whether the optimum location is moving or not.

### **3. Optimization of TMP Screening**

The target of the optimization is a refined rejects pressure screen in the thermomechanical pulping plant at the Bowater mill, located in Thunder Bay, ON. The screen's ability to separate long fibres from short fibres is to be optimized by manipulating the reject ratio and the shower water flow to feed flow ratio.

In this chapter, the current control scheme for the refined rejects pressure screen is described. The experimental results are compared against the predicted values. The process of determining the screening cost function is shown along with a breakdown for each segment of the equation. Also, computer simulations were run to test the ability of the equilateral optimum following simplex method and the non-equilateral optimum following simplex method to control the operation of the screen. Several illustrations show how each control method reacts to disturbances, and how the characteristics of the simplex can effect its ability to search out the optimum operating conditions.

#### **3.1 Reject Screen and Instrumentation**

The instrumentation associated with the refined rejects pressure screen include flow meters on the shower, accepts and rejects streams. The rejects consistency is measured by a consistency sensor. This project requires knowledge of the accepts fibre distribution and freeness, which is measured on-line using a PQM 400.

The PQM 400, Pulp Quality Monitor 400, is a measurement system for on-line measurement of the following pulp quality properties: drainage, fibre length and shive content. The sample analyzed by the PQM 400 is extracted from the accepts stream exiting the screen. The PQM 400 is designed so that samples can be analyzed from four separate streams throughout the plant. The process of sampling, analysis, monitoring of streams and discharge from the PQM 400 takes ten minutes to complete, therefore, using the current setup, a measurement of the accepts stream from the screen is only available every 40 minutes [18].

The drainage time of the sample is converted to a freeness value in Canadian Standard Freeness (CSF). The fibre length of the sample is measured and calculated in three categories: fine, medium and coarse fractions [18]. The range of each size category is listed in Table 3.1. Although the shive content of

the sample can be measured by the PQM 400, the shive content was not selected as a variable to optimize in this project.

A diagram of the control structure for the refined rejects pressure screen is shown in Figure 3.1. The loop diagram for the differential pressure is shown in Figure 3.2. Typically, the feed pressure is 175 kPa and the accepts pressure is 235 kPa. The ratio of the shower flow rate to the feed flow rate determines the shower flow. The valve on the shower stream has a range of 0 to 2500L/min. The typical flow rate is 650L/min, and the ratio of shower to feed is 5.5%. The loop diagram for the shower flow control is shown in Figure 3.3. The reject flow rate is controlled similarly to the shower flow. The setpoint is determined by the volumetric reject ratio, usually 14%. The valve on the reject stream has a range of 0 to 2500L/min. The typical reject flow rate is 1750L/min. The loop diagram for the reject flow control is shown in Figure 3.4.

PQM Fibre Size	Range of Length (mm)	Percentage of Fibre in Feed
Fine	0.10-0.35	26.1%
Medium	0.35-1.95	56.2%
Coarse	1.95-7.00	17.7%

Table 3.1: Fibre size categories and ranges of length.

### 3.2 Experimental Data

Modelling a pressure screen involves knowledge of the overall process along with the internal design. Knowledge of the process begins with the feed stream. Typically, the feed flow rate is 12000 L/min, with a consistency of 2.4%, and a freeness of 204 CSF. A fibre size population distribution curve of a sample from the feed stream is shown in Figure 3.5 which is the average of five samples taken from the feed stream on the refined rejects pressure screen.

The feed fibre size distribution shown in Figure 3.5 can be classified into the three groups, and the percentage of each is listed in Table 3.1 [18]. Although the coarse fibre category extends to 7mm, Figure 3.5 shows that very little fibre is longer than 4mm.

Figure 3.6 illustrates the experimental passage ratio along with a line drawn using Equation (1.21). the line shown in Figure 3.6 was determined using trial and error. Several values of  $\lambda$  were tried in Equation (1.21) until the value of 6.67 was estimated to best fit the data.

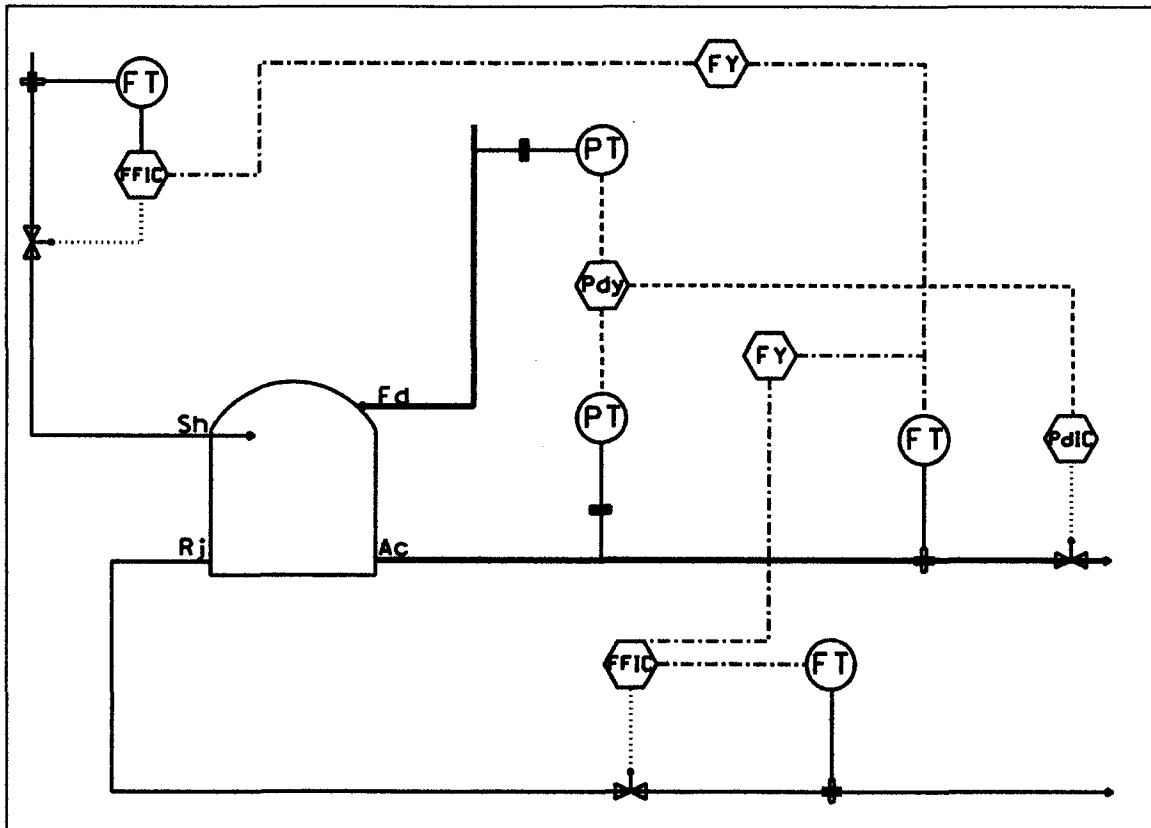


Figure 3.1. A diagram of the reject pressure screen and its control structure.

Figure 3.2. Instrument loop diagram for the differential pressure control across the screen





$$P(l) = e^{-l/\lambda} \quad (1.21)$$

For the plug-flow method, the predicted rejects fibre distribution was calculated using Equations (1.18) and (1.21). The accepts fibre distribution was calculated using Equations (1.23) through (1.25). The value of the fibre coarseness,  $\sigma$ , in Equation 1.23 was not available through measurement and was assumed to be one. The distributions are categorized based on the range of length in Table 3.1, and are listed in Table 3.2.

$$c_r(l) = R_v^{P(l)-1} c_f(l) \quad (1.18)$$

$$m_r(l) = c_r(l) Q_r l \sigma \quad (1.23)$$

$$m_a(l) = m_f(l) - m_r(l) \quad (1.24)$$

$$c_a(l) = \frac{m_a(l)}{Q_a \rho_a} \quad (1.25)$$

Fibre Size	Accepts (% of fibre)	Rejects (% of fibre)
Fine	27.4%	20.0%
Medium	56.6%	54.4%
Coarse	16.0%	25.6%

Table 3.2. Fibre size distribution for accepts and rejects using the plug flow method.

Typically, the consistency of the rejects stream is about 3.46% (mass of dry fibres/mass of stock). Using Equation (1.30), the predicted reject consistency is 3.27%.

$$C_r = \sum_{l=0}^{7mm} c_r(l) \quad (1.30)$$

For the mixed-flow method, the rejects fibre population distribution was calculated using Equations (1.21) and (1.36). The accepts fibre distribution was calculated using Equation (1.34). Both of the distributions were summed into the size categories, and are listed in Table 3.3

$$c_r(l) = \frac{1}{P(l) - P(l)R_v + R_v} c_f(l) \quad (1.36)$$

$$c_a(l) = P(l) c_r(l) \quad (1.34)$$



Fibre Size	Accepts (% of fibre)	Rejects (% of fibre)
Fine	27.0%	21.4%
Medium	56.5%	54.6%
Coarse	16.5%	24.0%

Table 3.3. Fibre size distribution for accepts and rejects using the mixed-flow method.

The predicted consistency for the mixed-flow method was 2.81%, compared to the plug-flow prediction of 3.07%, and the 3.46% experimental consistency. Neither of the methods of predicting the rejects consistency proved to be very close to the experimental value.

Based on the comparison of fibre distributions and the consistency predictions, the plug flow method is the better choice when attempting to model the screen. While the mixed-flow method is acceptable, the plug-flow method is better at predicting the consistency of the reject stream [8].

Fibre Size	Accepts (% of fibre)	Rejects (% of fibre)
Fine	27.0%	22.3%
Medium	56.8%	53.7%
Coarse	16.2%	24.0%

Table 3.4. Fibre size distribution for accepts and rejects from experimental data.

With a lack of literature on pulp freeness as a function of the fibre size distribution, the only guideline to use is that freeness decreases as the percentage of fines increase. Using this guideline an equation describing accepts freeness ( $F_a$ ) can be created, and is shown in Equation (3.1).

$$F_a = -3.223 * FIN_a + 253.19 \quad (3.1)$$

where  $F_a$  is the freeness of the accepts in CSF.

$FIN_a$  is the fine fibre fraction for the accepts stream.

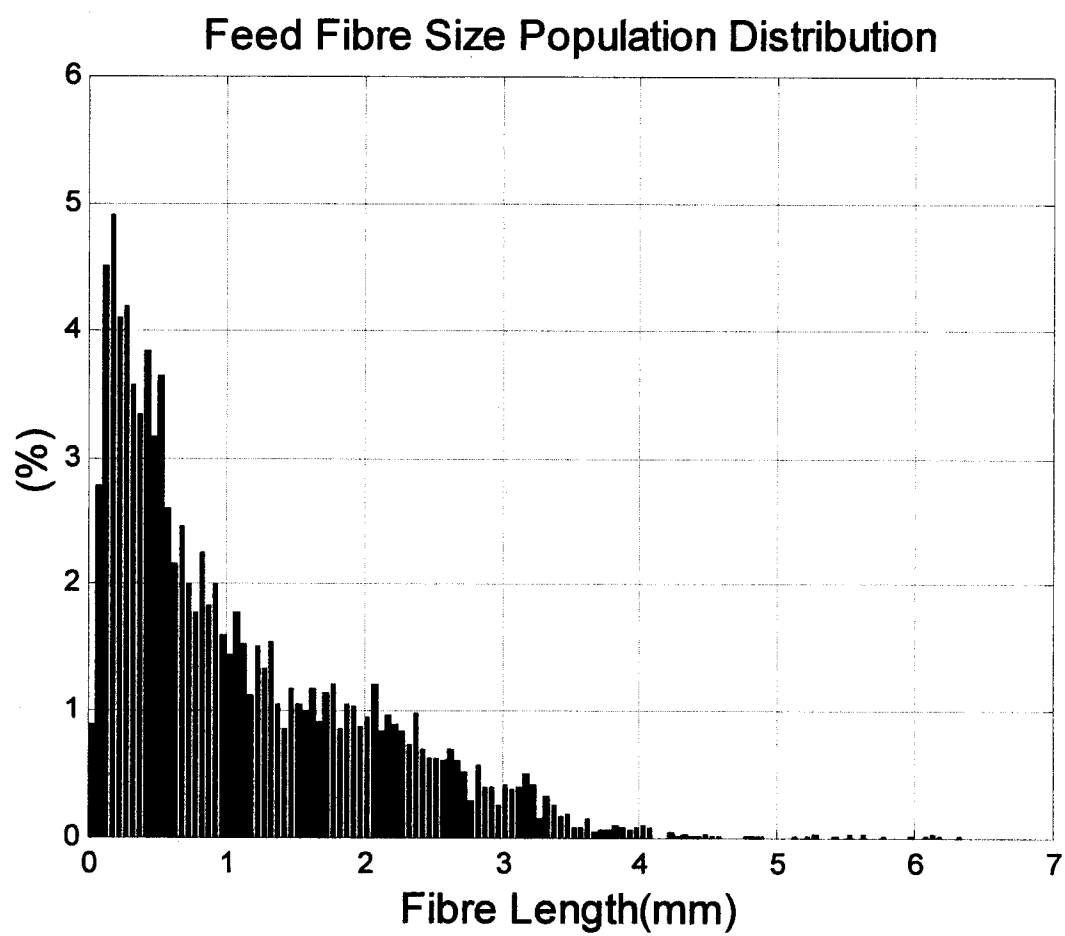


Figure 3.5. Experimental feed fibre size distribution.

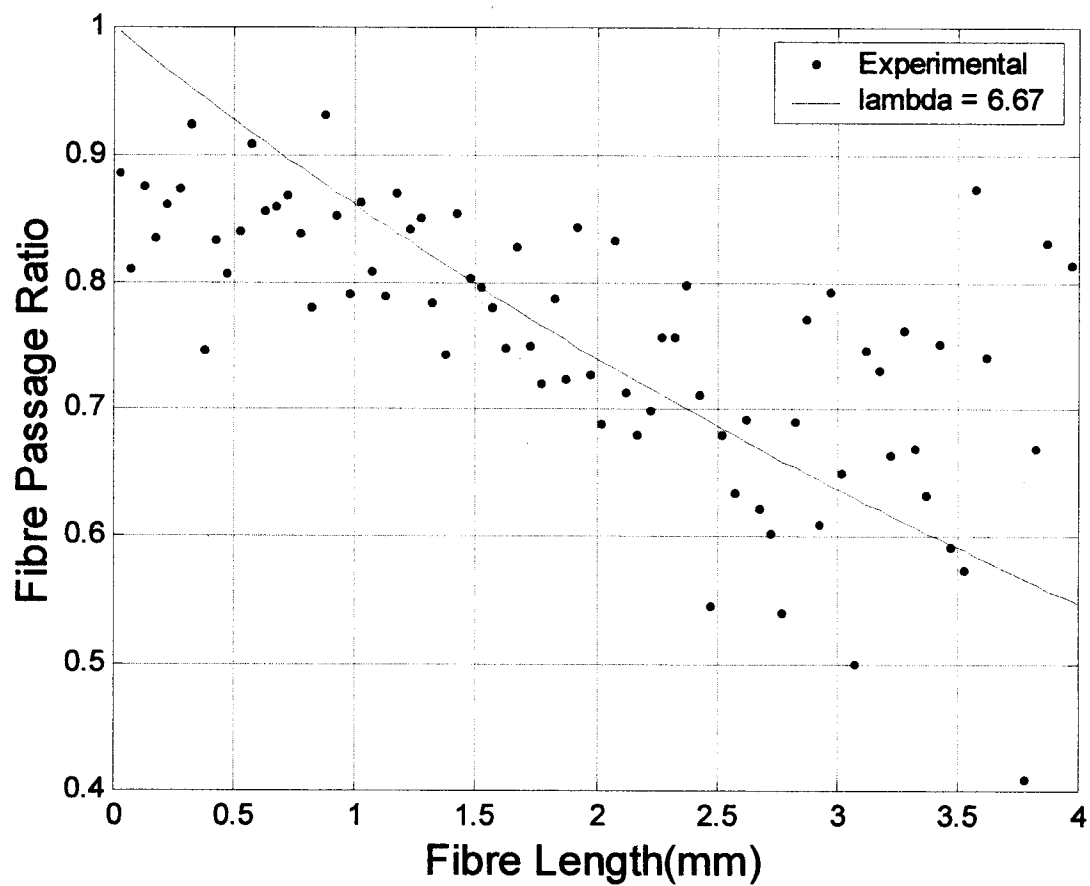


Figure 3.6. Fibre passage ratio calculated from experimental data.

### 3.3 The screening cost function

Optimizing the refined reject pressure screen involves the specification of a performance equation of controlled and the selection of manipulated variables. The performance equation will involve variables like fibre removal efficiency, accepts freeness and rejects consistency. Alternatively, fibre removal efficiency can be replaced by a specified fibre size distribution if that is the goal of the screening. The manipulated variables are the volumetric reject ratio, listed in Equation (1.17), and the shower flow to feed flow ratio, shown in Equation (3.2).

$$R_v = \frac{Q_r}{Q_f + Q_s} \quad (1.17)$$

$$R_s = \frac{Q_s}{Q_f} \quad (3.2)$$

If the objective function is to be based on fibre removal efficiency then the performance equation can include the long ( $e_{lg}$ ) and short ( $e_{sh}$ ) fibre removal efficiencies. The absolute value of the difference between them can be used to determine a performance equation, given in Equation (3.3), where  $K_1$  is a constant to keep the performance value positive.

$$perf = -|e_{lg} - e_{sh}| + K_1 \quad (3.3)$$

The goal is to maximize the difference between the two efficiencies. Since each of the fibre removal efficiencies have values between zero and one, perfect separation is signaled by an absolute value of one, while no separation is indicated by an absolute value of zero. A surface plot of Equation (3.3) is shown in Figure 3.7 for a range of shower and reject ratio's. The fibre separation cost equation is influenced by the reject ratio far more than the shower ratio. The range of shower and reject ratio's chosen were based on the minimum and maximum values used during experimental trials at Bowater's, Thunder Bay TMP facility. The greater the reject ratio, the better the fibre separation. Examining Figure 3.7, the shower ratio has little effect on the fibre separation, largely due to the small flow of shower water relative to the feed stream.

The alternative performance equation uses the three fibre fractions provided by the PQM 400 as shown in Equation (3.4).

$$perf = \omega_1(x_{fin} - x_{fin_{TAR}})^2 + \omega_2(x_{med} - x_{med_{TAR}})^2 + \omega_3(x_{cor} - x_{cor_{TAR}})^2 \quad (3.4)$$

In Equation (3.4),  $\omega_1, \omega_2, \omega_3$  are weighting factors applied to the square of the errors of each fibre size category, while  $x_{fin}, x_{med}, x_{cor}$  are the measured values provided by the PQM 400 for the fine, medium and coarse fibre fractions, respectively. The  $x_{fin_{TAR}}, x_{med_{TAR}}, x_{cor_{TAR}}$  variables in Equation (3.4) represent the target fibre fractions chosen by the operator for the fine, medium and coarse fibre fractions.

- $\omega_1 = 1.0, \omega_2 = 1.0, \omega_3 = 1.0$
- $x_{fin_{TAR}} = 26.4\%, x_{med_{TAR}} = 56.0\%, x_{cor_{TAR}} = 17.6\%$

The surface plot for Equation (3.4) is illustrated in Figure 3.8. Similar to the maximum separation cost equation, the optimum reject ratio is at the upper end of the range tested, while the shower ratio has little effect on the distribution. Based on both the fibre separation and target distribution surface plots, the optimum reject ratio is as large as the screen is capable of handling.

The accepts freeness has a target value ( $F_{a_{TAR}}$ ) that provides the best pulp properties for paper making. Using the square of the error between the measured freeness ( $F_a$ ) and the target freeness, and multiplying it by a weighting factor ( $\omega_4$ ), the performance equation penalizes any value less than or greater than the target value, as shown in Equation (3.5). The value of the weighting factor is 0.03125 and the target freeness is 175CSF. The surface plot of Equation (3.5) is shown in Figure 3.9.

$$perf = \omega_4(F_a - F_{a_{TAR}})^2 \quad (3.5)$$

The optimum reject ratio for the freeness cost equation is just below 0.1. Once again the shower ratio has little effect on the cost equation for the accepts freeness. The insensitivity towards a change in shower ratio exists because it has very little effect on the freeness of the pulp. Increasing the flow of shower water in the screen doesn't alter the drainage properties of the pulp. The drainage of the pulp is altered by the percentage of fines. As the percentage of fine fibres increase within the pulp, the freeness decreases because the small fibres have the ability to plug the gaps between the larger fibres.

A reject consistency too large can cause screen plugging. Although a low consistency doesn't hamper screen operation, it is an indication that an excessive amount of shower water is being used. The

use of boundary constraints for the rejects consistency in the cost equation will penalize any reject consistency that deviates from the target value. The use of a hyperbolic tangent equation instead of a square of the error equation allows for control of each side of the target value, independently. The height and slope of the penalty function can be given different values on each side of the target rejects consistency.

$$perf = \omega_5 [k_1 + \tanh(k_2 (C_r - C_{r_{\max}}))] - \omega_6 [k_3 + \tanh(k_4 (C_r - C_{r_{\min}}))] + K_2 \quad (3.6)$$

An equation using hyperbolic tangent can be used to increase the penalty as the reject consistency ( $C_r$ ) approaches and exceeds a maximum allowable consistency ( $C_{r_{\max}}$ ), and is shown in Equation (3.6) [20].

$\omega_5$  and  $\omega_6$  is a weighting factor applied to Equation (3.6),  $k_1$  and  $k_3$  controls the overall height of the penalty,  $k_2$  and  $k_4$  control the slope of the penalty function, and  $K_2$  is a constant to ensure the equation remains positive.

- $\omega_5 = 1.0, k_1 = 1.0, k_2 = 6.0, \omega_6 = 3.5, k_3 = 1.0, k_4 = 3.0$
- $C_{r_{\max}} = 3.50\%, C_{r_{\min}} = 2.70\%, K_2 = 0.95$

Another reason for penalizing the low rejects consistency is that the optimum shower ratio was found to be as large as the valve throughput could provide. Allison and Olson also found that screen runability improves with increased shower water, but shower flow must be limited due to increased dewatering costs [9]. In Figure 3.10, the surface plot for the rejects consistency is illustrated. The penalty for the low consistency is much steeper than the high consistency to imply strict limits on the maximum shower and reject ratio's. A valley defines the minimum of the surface plot in Figure 3.10.

The addition of Equations (3.3), (3.5) and (3.6) creates a surface equation in which a global minimum exists. The combined equations are illustrated in the surface plot shown in Figure 3.11, and the contour plot in Figure 3.12. A problem with the cost equation illustrated in Figure 3.11 and 3.12 is the shallow trough that exists for a large range of reject ratios and all shower ratios. This is a problem because the valve position for either the reject or shower streams may approach a fully opened or closed position too easily.

To prevent the possibility that the minimum may lie too close to 0 or 100% for either the reject or shower flow valve positions, four boundaries were added for the valves using Equation (3.7).

$$perf = \omega_7[k_5 + \tanh(k_6(R_s - R_{s_{ll}})))] + \omega_8[k_7 + \tanh(k_8(R_v - R_{v_{ll}})))] \\ + \omega_9[k_9 + \tanh(k_{10}(R_s - R_{s_{ul}})))] + \omega_{10}[k_{11} + \tanh(k_{12}(R_v - R_{v_{ul}})))] \quad (3.7)$$

where,  $\omega_7, \omega_8, \omega_9, \omega_{10}$  are weighting factors for each of the ratio limits.

$k_5, k_7, k_9, k_{11}$  control the overall height of each of the  $\tanh$  functions.

$k_6, k_8, k_{10}, k_{12}$  are the slope of each of the  $\tanh$  functions.

$R_s$  is the current shower to feed ratio.

$R_{s_{ll}}, R_{s_{ul}}$  are the lower and upper limits for the shower to feed ratio.

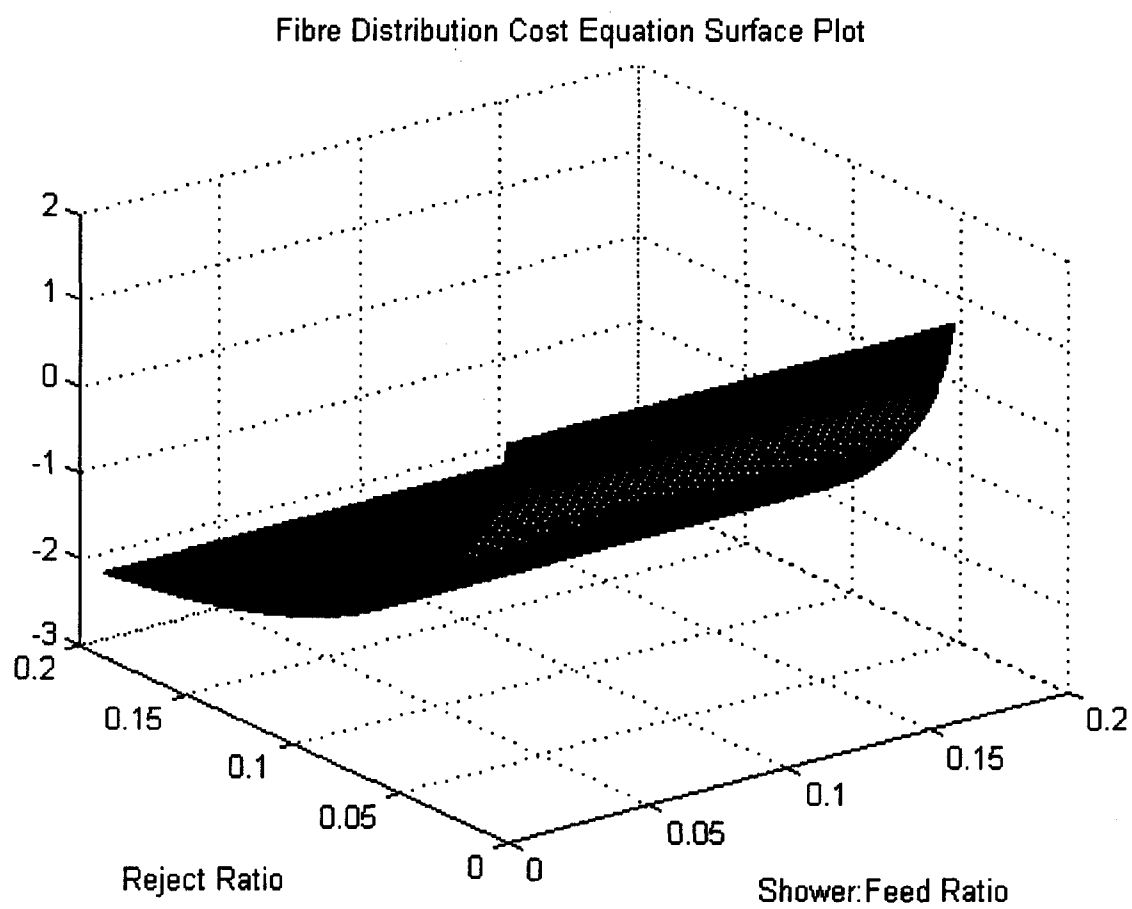
$R_v$  is the current volumetric reject ratio.

$R_{v_{ll}}, R_{v_{ul}}$  are the lower and upper limits for the volumetric reject ratio.

- $\omega_7 = -1.0, k_5 = 1.0, k_6 = 100, R_{s_{ll}} = 0.15$
- $\omega_8 = -0.5, k_7 = 1.0, k_8 = 100, R_{v_{ll}} = 0.04$
- $\omega_9 = 1.5, k_9 = 1.0, k_{10} = 75, R_{s_{ul}} = 0.15$
- $\omega_{10} = 1.0, k_{11} = 1.0, k_{12} = 75, R_{v_{ul}} = 0.19$

The surface plot of Equation (3.7) is displayed in Figure 3.13. Each of the walls in the surface plot is near the minimum or maximum ratio to prevent the simplex from taking the valves to 0 or 100% open.

Adding Equation (3.7) to Equations (3.3), (3.5) and (3.7) results in the overall cost equation, and the surface plot illustrated in Figure 3.14. The contour plot of the overall cost equation is shown in Figure 3.15. It shows the effect the boundaries on the valves have on the overall cost equation. There now exists a slope at each end of the shower ratio range that will penalize any that approach to these extremes. The minimum of the surface plot in Figure 3.14 and 3.15 is located at 0.095 for the shower to feed ratio, and 0.13 for the reject ratio.



**Figure 3.7. Surface plot of the fibre separation cost equation.**



### Target Fibre Distribution Cost Equation Surface Plot

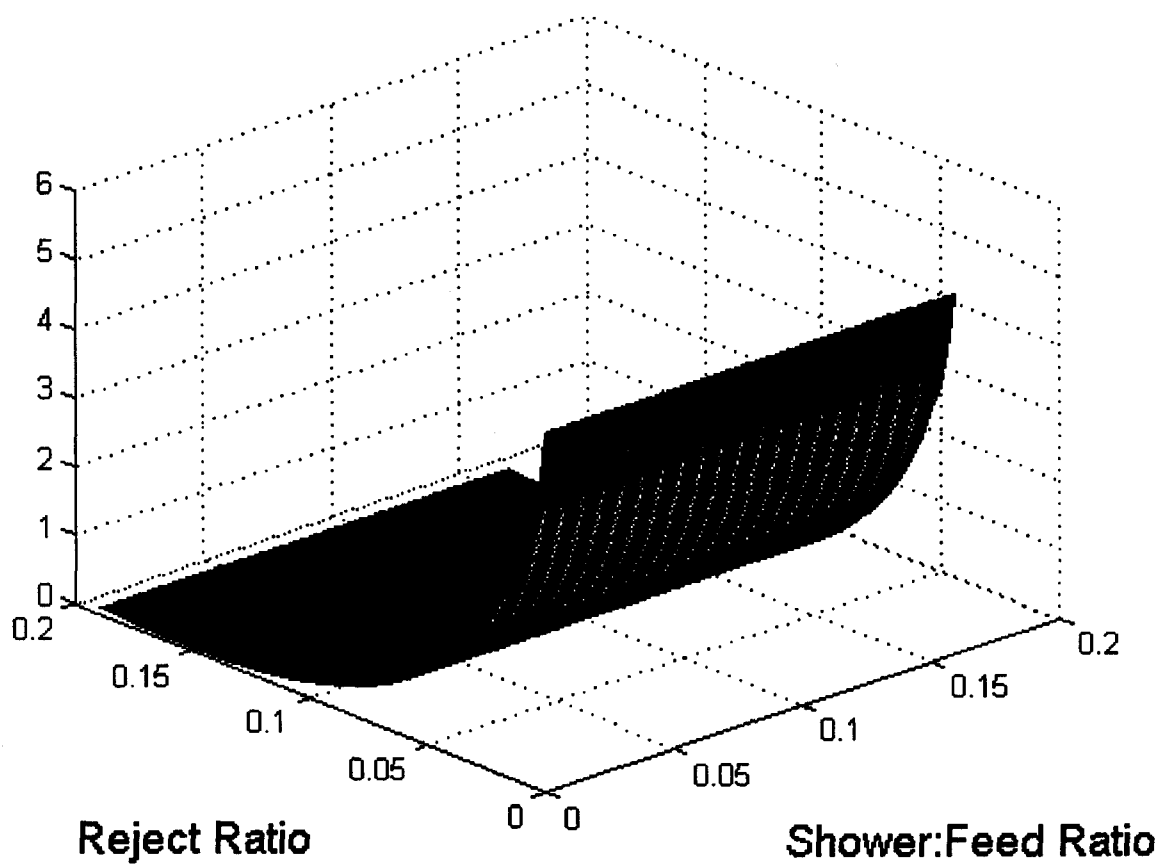


Figure 3.8. Cost surface for the target fibre distribution equation.

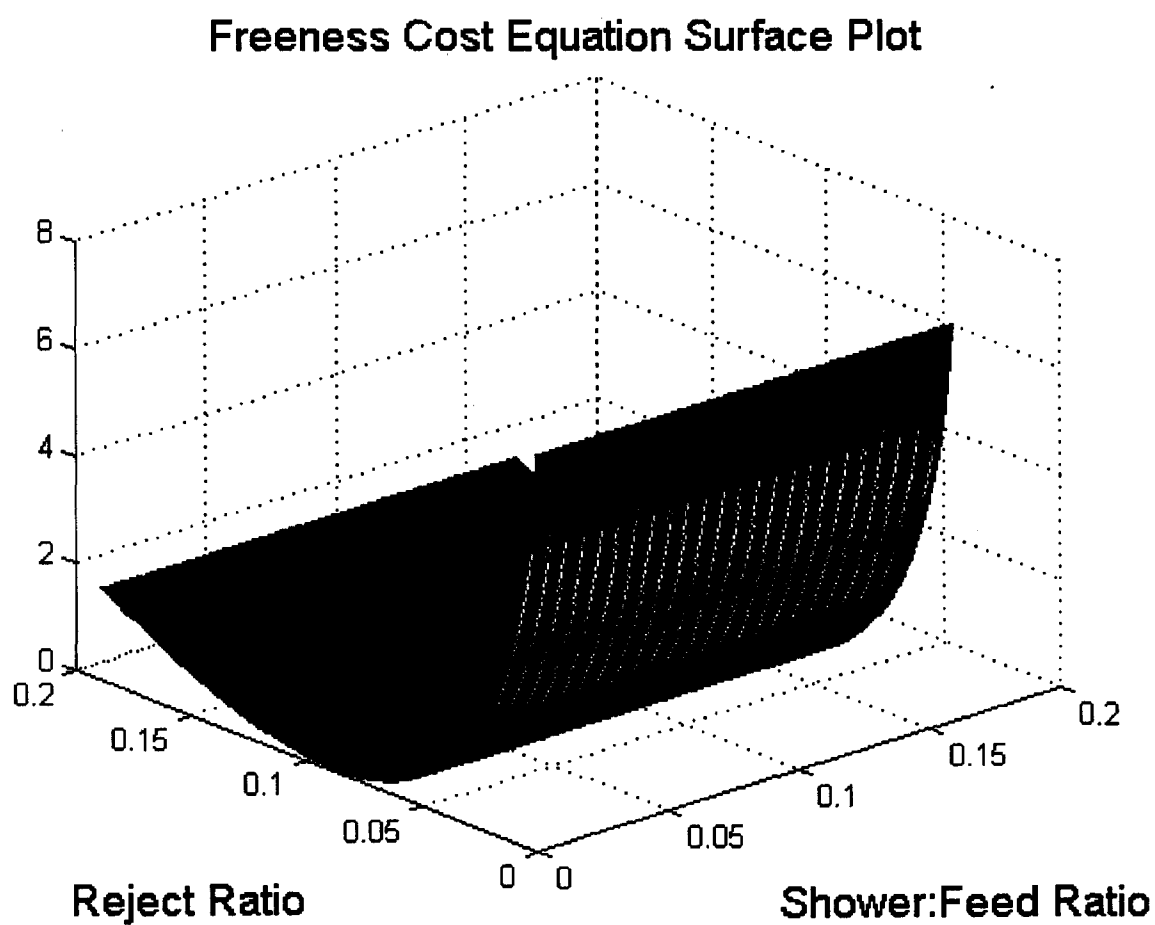


Figure 3.9. Cost surface for the accepts freeness equation.

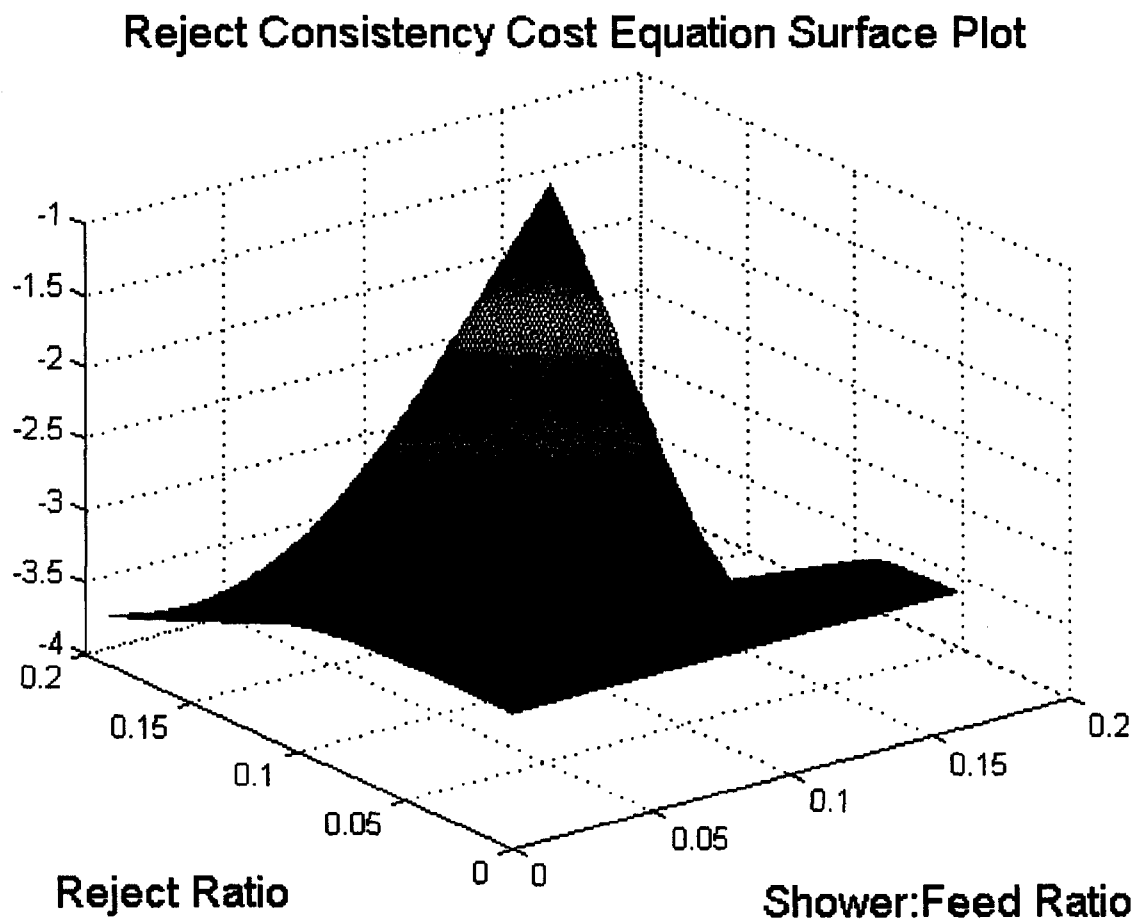


Figure 3.10. Cost surface for the rejects consistency.

### Cost Equation Surface Plot without Valve Constraints

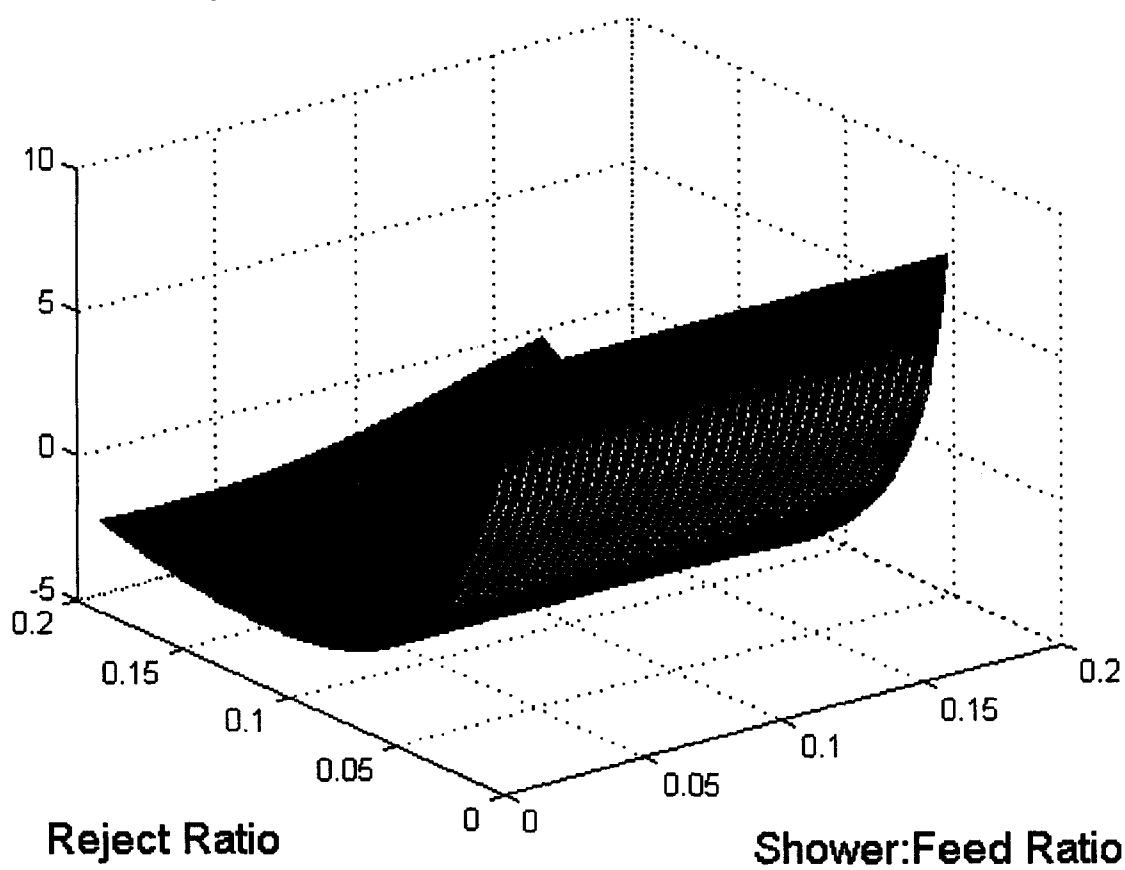


Figure 3.11. Surface for the overall cost equation without boundaries on the valves.

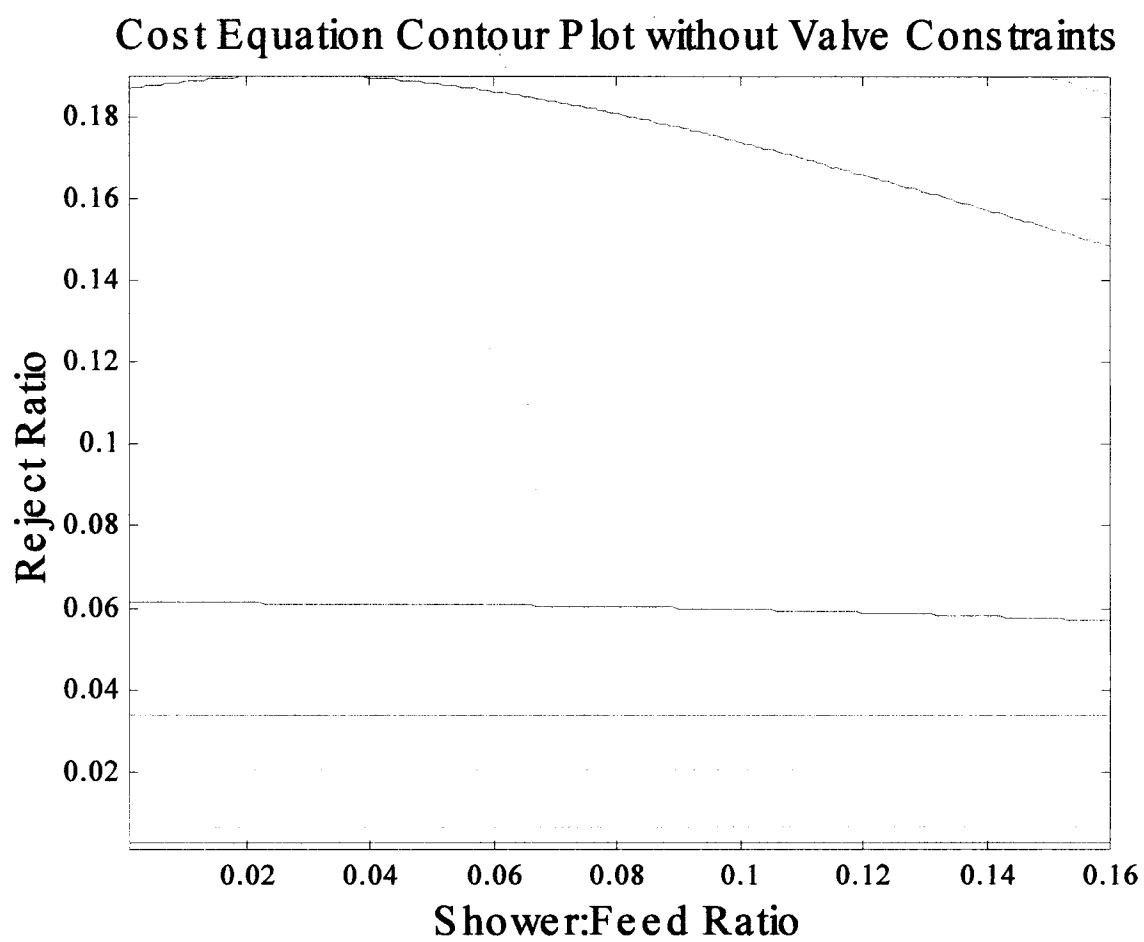


Figure 3.12. Contour plot of the overall cost equation without the valve boundaries.

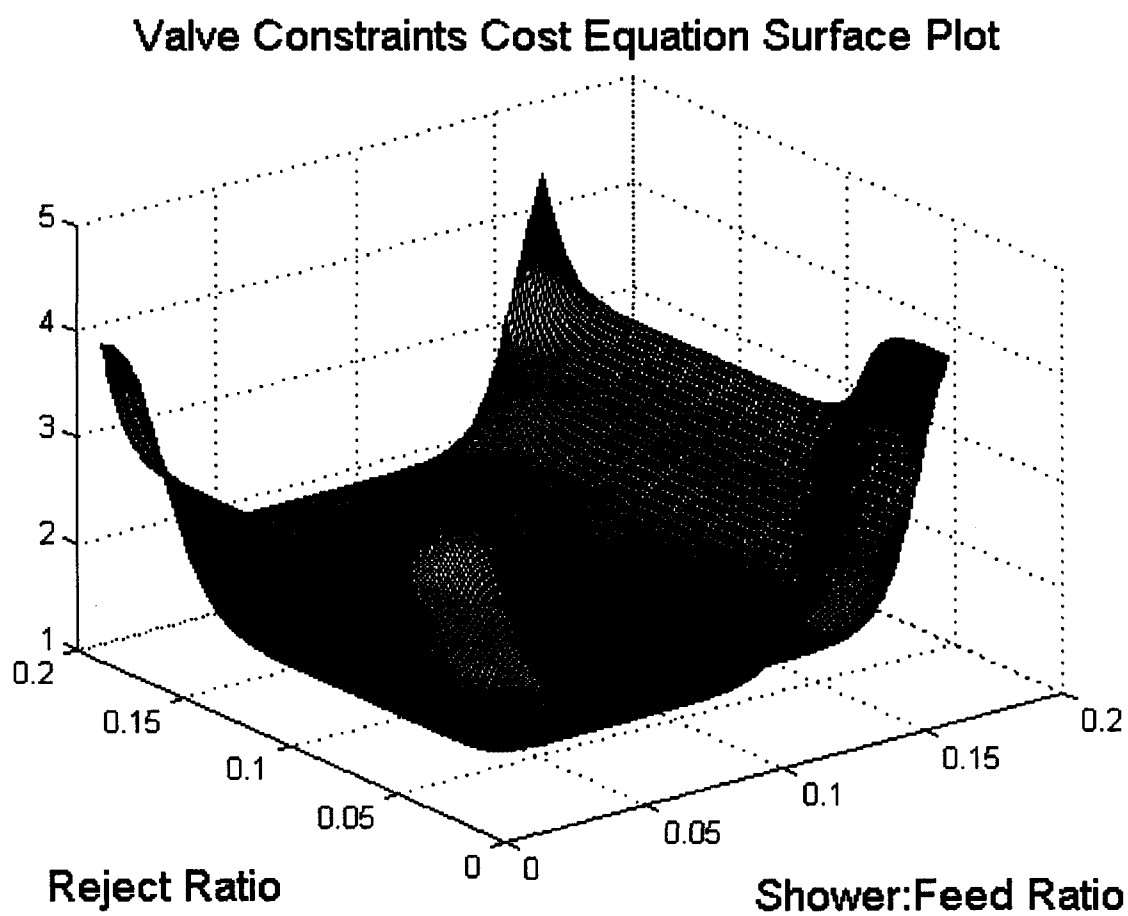


Figure 3.13. Cost surface for the valve boundaries.

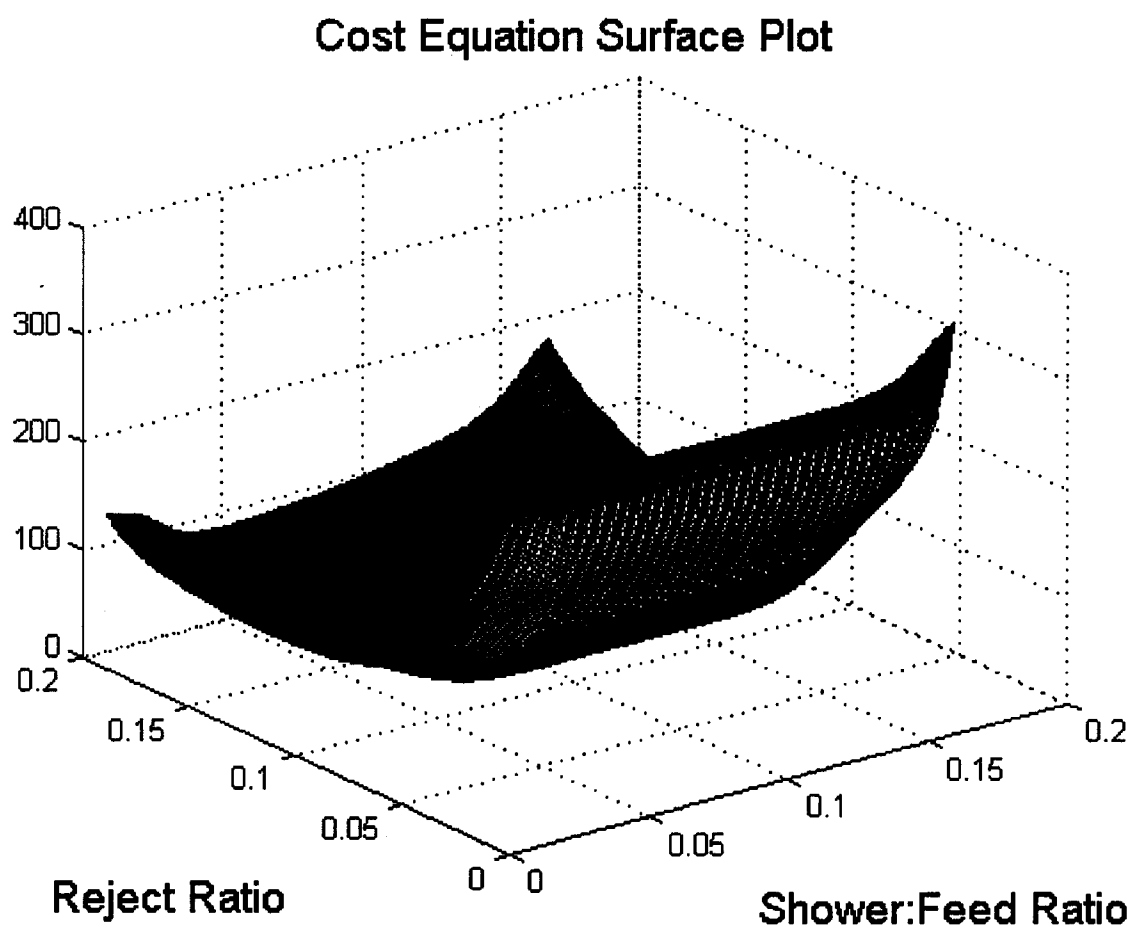


Figure 3.14. Surface for the overall cost equation.

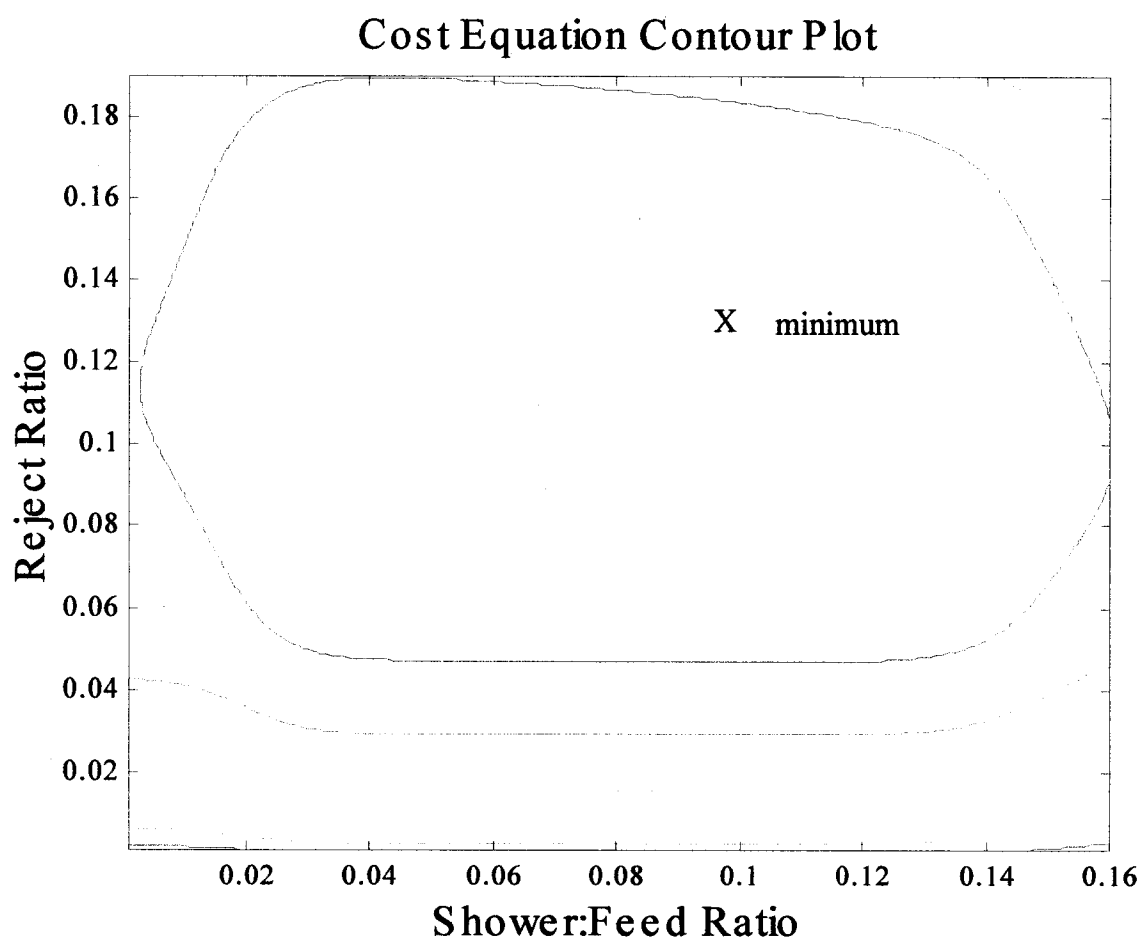


Figure 3.15. Contour plot for the overall cost equation.



### 3.4 Reject Screen Optimization

Once both of the simplex methods have been tested on a known optimum, they can be applied to the simulation of the pressure screen. Table 3.5 lists the various simulation combinations using the EOFS method. The columns of Table 3.5 show how the EOFS method works using a variety of expansion and contraction factors. The final column includes noise, which should indicate how the optimization method works in real conditions where the analysis of the samples will most likely be noisy. The rows of Table 3.5 list the different disturbances applied to the simulation.

In Chapter Two it was shown that both the EOFS and NOFS methods work effectively when the process is free from disturbances, but both must be tested against several disturbances to prove its overall effectiveness and potential in an industrial setting. The simplex can be tested against disturbances in feed flow, feed fibre distribution and feed consistency. Some of these disturbances will cause a shift in the location of the optimum. Simulations of the process with the disturbances will determine the usefulness of the simplex as a technique in optimizing a pulp pressure screen.

The first type of disturbance to be simulated is feed flow. The feed flow disturbance is simulated in EOFS trials one through three. Changes to the feed flow rate shouldn't cause any problems to the simplex because the variables being controlled are ratio's, and not flows. Any change in the feed flow rate should cause the shower and reject ratio's to remain constant, only changing the flow rates. The shower flow rate needs to change at the same rate as the feed flow to prevent the screen from plugging if the feed flow increases, or to prevent excessive watering if the feed flow decreases. The reject flow should also change at the same rate as the feed flow to keep the output of the screen constant.

The simulation shown in Figures 3.16(a) and (b) is from Trial EOFS #1. The starting feed flow rate was 11500L/min. A 2000L/min feed flow disturbance was added to the simulation after 25 samples. After 50 samples, a -4000L/min disturbance was entered into the simulation. Disturbances were continually added in the form of a square sine wave for the duration of the simulation. Some of the operating variables for the EOFS method simulations shown in Figures 3.16 through 3.18 are as follows:

- starting simplex size = 0.0025
- minimum simplex size = 0.0002
- maximum simplex size = 0.0045

<b>EOFS Method</b>	Expands 2.0 Contracts 0.5 <i>(maximizes fibre separation)</i>	Expands 1.25 Contracts 0.8 <i>(maximizes fibre separation)</i>	Expands 2.0 Contracts 0.5 With Noise <i>(maximizes fibre separation)</i>
Feed Flow Disturbance	EOFS #1	EOFS #2	EOFS #3
Feed Distribution Disturbance		EOFS #4	EOFS #5
Feed Consistency Disturbance		EOFS #6	EOFS #7

Table 3.5: Combinations of the conditions used to create the simulations for the EOFS method.

<b>NOFS Method</b>	No Ratio for Exp./Cont. <i>(maximizes fibre separation)</i>	Exp./Cont. Ratio 1.7 <i>(maximizes fibre separation)</i>	Exp./Cont. Ratio 1.7 <i>(target fibre distribution cost equation)</i>	Exp./Cont. Ratio 1.7 with Noise <i>(maximizes fibre separation)</i>
Feed Flow Disturbance	NOFS #1	NOFS #2	NOFS #3	NOFS #4
Feed Distribution Disturbance	NOFS #5	NOFS #6	NOFS #7	NOFS #8
Feed Consistency Disturbance	NOFS #9	NOFS #10	NOFS #11	NOFS #12

Table 3.6: Combinations of the conditions used to create the simulations for the NOFS method.

The initial ratio's were chosen to be in the range of the normal operating values of the screen at Bowater. The starting reject ratio was 0.138 and the shower ratio was 0.055. The 2000L/min feed flow disturbance has little or no effect on the operation of the simplex. Trial EOFS #1 uses an expansion factor of 2 and a contraction factor of 0.5, therefore the simplex doesn't settle down well, and the reject and shower ratio's oscillate. An expansion factor of 2 and a contraction factor of 0.5 can lead to substantial changes in the simplex size. Each time the simplex expands, it doubles in size.

The first two charts of Figure 3.16(b) show the trends of the reject and shower ratios. The abbreviation, *rjrat*, represents the reject ratio of the process, *optrj*, represents the optimum reject ratio, *shrat*, represents the shower ratio, and, *optsh*, represents the optimum shower ratio.

The reject ratio is oscillating in a range just below the estimated optimum based on the cost equation. The shower to feed ratio is oscillating just below the estimated optimum for the first 50 samples. At that point the simplex begins to increase the value of the shower to feed ratio. This increase is due to a combination of the simplex pivoting about a single point and the size of the simplex at that time. During that time period the simplex is large enough that the shower ratio oscillates above the optimum. Over the next several samples, the simplex continues to shrink and expand in an area just greater than the shower optimum. This simulation could be improved by using smaller expansion and contraction steps that would lead to smaller oscillations. When any of the feed flow disturbances enter the simulation, neither of the reject or shower ratios are affected.

The size of the oscillations for the simulation shown in Figures 3.17(a) and (b) are for Trial EOFS #2 are smaller than those in the simulation Trial EOFS #1. The expansion factor was reduced from 2.0 to 1.25, and the contraction factor was changed from 0.5 to 0.8. The smaller oscillations are a result of the smaller expansion and contraction factors. Although the expansion and contraction factors are changed, the maximum and minimum simplex size is unchanged. This is evident during the first 40 samples of the simulation where the oscillations are as large as those in Trial EOFS #1. To further tighten the size of the oscillations the maximum simplex size could be reduced.

Figures 3.18(a) and (b) are illustrations of the simulation Trial EOFS #3. This simulation has noise added to the measurement of the fibre distribution and the accepts and rejects consistency. The use of this method worked well before and after the disturbances enters the process for each of the two previous

simulations. In this simulation the noise must be disrupting the EOFS method from closing in on the optimum reject ratio. This result is something that must be observed during other simulations that include noise.

Some of the operating variables for the NOFS method simulations shown in Figures 3.19 through 3.22 are as follows:

- starting simplex size = 0.005
- minimum simplex size = 0.0002
- maximum simplex size = 0.005

The use of the NOFS method against a 2000L/min feed flow disturbance is shown in the simulations shown in Figures 3.19 through 3.22. In the third column of simulations in Table 3.6, the performance equation used is the target fibre distribution. For all other results in Tables 3.5 and 3.6, the performance equation is based on maximizing the fibre separation. For all three simulations using the target fibre distribution performance equation, the accepts stream target fibre distribution is listed below:

- Accepts Target Fine Fibres = 27.9%
- Accepts Target Medium Fibres = 57.4%
- Accepts Target Coarse Fibres = 14.7

The simulation shown in Figures 3.19(a) and (b) is from Trial NOFS #1. In Figures 3.19(a) and (b), the simulation uses the simplex without the use of a ratio for the expansion or contraction. Although no constraints were placed on the shape of the simplex, it did not become elongated in one direction. The changes for one of the variables may have become much greater than the other variable without constraints on the simplex shape. In Figure 3.19(b), the simplex searches out the optimum reject and shower ratios in less than 20 samples. After finding the optimum, the simplex stays very close to the optimum by shrinking and oscillating just below the optimum reject ratio and shower ratio. This simulation seems to work very well.

Trial NOFS #2 is illustrated in Figures 3.20(a) and (b). This simulation uses a constraint on the shape of the simplex. The constraint prevents the simplex from becoming too elongated. It ensures that the distance from the centroid to the furthest vertex is not more than 1.7 times greater than the distance from

the centroid to the nearest vertex. The results of Trial NOFS #2 are similar to those illustrated in Trial NOFS #1.

Trial NOFS #3 is illustrated in Figures 3.21(a) and (b). This simulation uses a constraint on the shape of the simplex and replaces the maximum fibre separation term with three target fibre distribution terms. The simplex searches and maintains the optimum shower ratio very well. Unfortunately, the simplex is not as successful in searching for the optimum reject ratio. During the entire simulation the reject ratio stays a consistent distance away from the optimum. This distance may be a result of a very shallow slope on the performance equation surface near the optimum ratios.

Trial NOFS #4 is illustrated in Figures 3.22(a) and (b). This simulation uses the shape constraint along with noise added to the reject and accept consistencies and the accepts freeness. Considering the noise added, the simulation results are good because they show no signs of being influenced by the feed optimum reject ratio. The simplex oscillates just above the optimum shower ratio, but stays well below the flow disturbance, while still settling down as time passes. Similar to Trial EOFS #3, the search for the optimum reject ratio remains just below the calculated optimum.

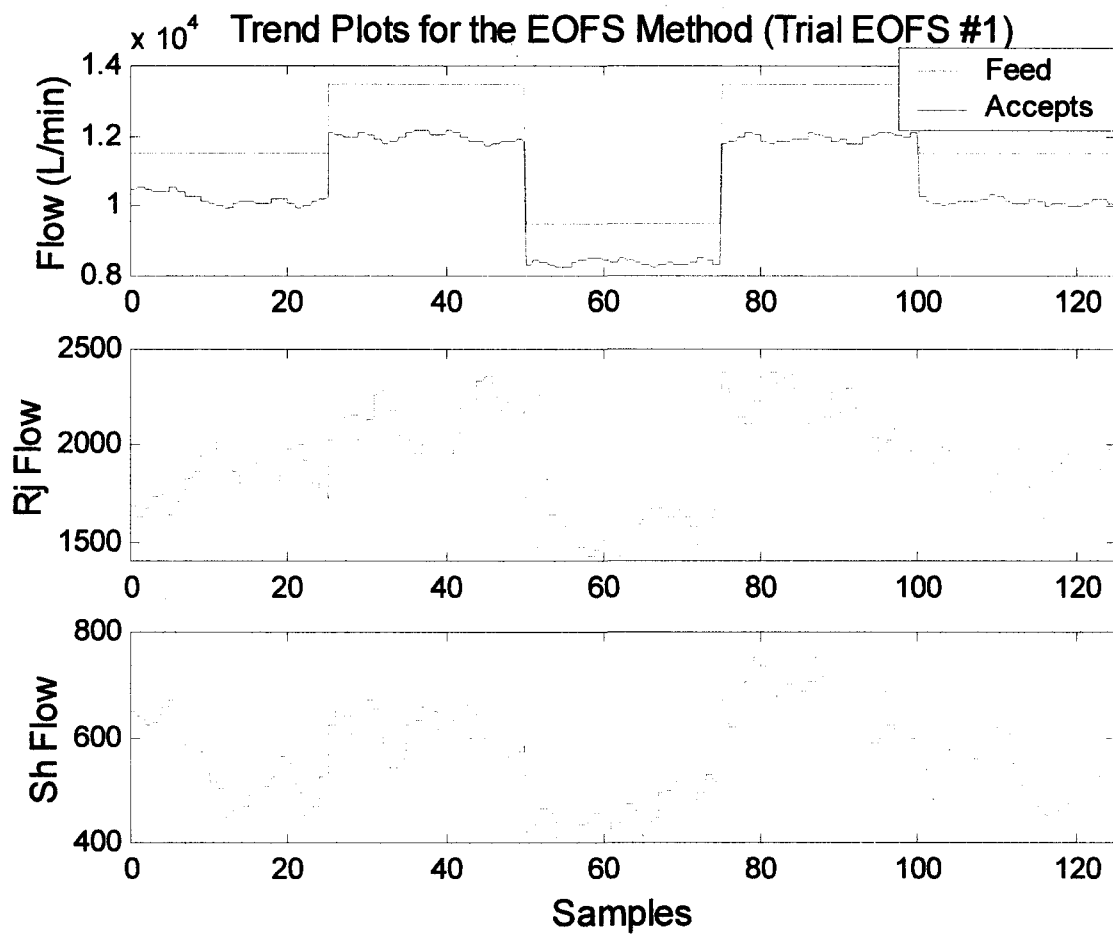


Figure 3.16(a): Trend plots for the flows of each stream for the simulation Trial EOFS #1.

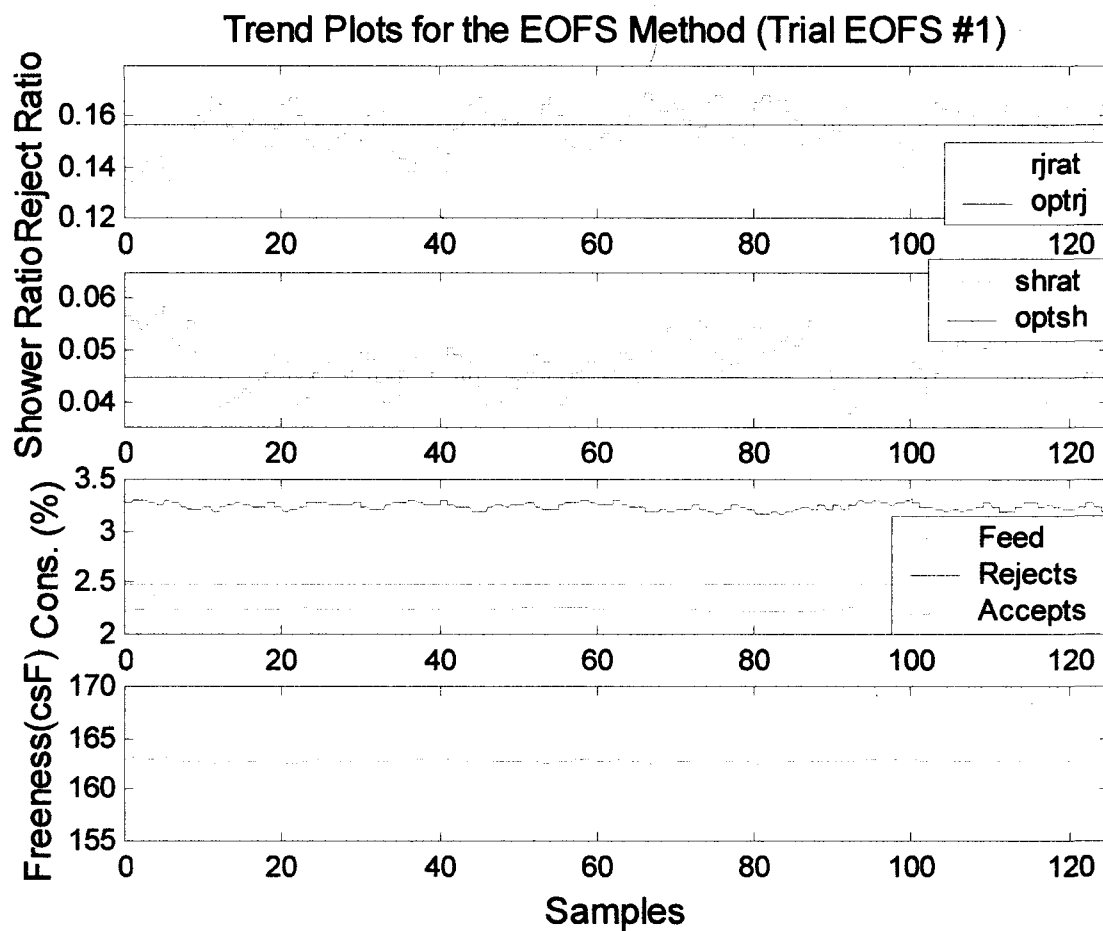


Figure 3.16(b): Trend plots for the operating variables for the simulation Trial EOFS #1.

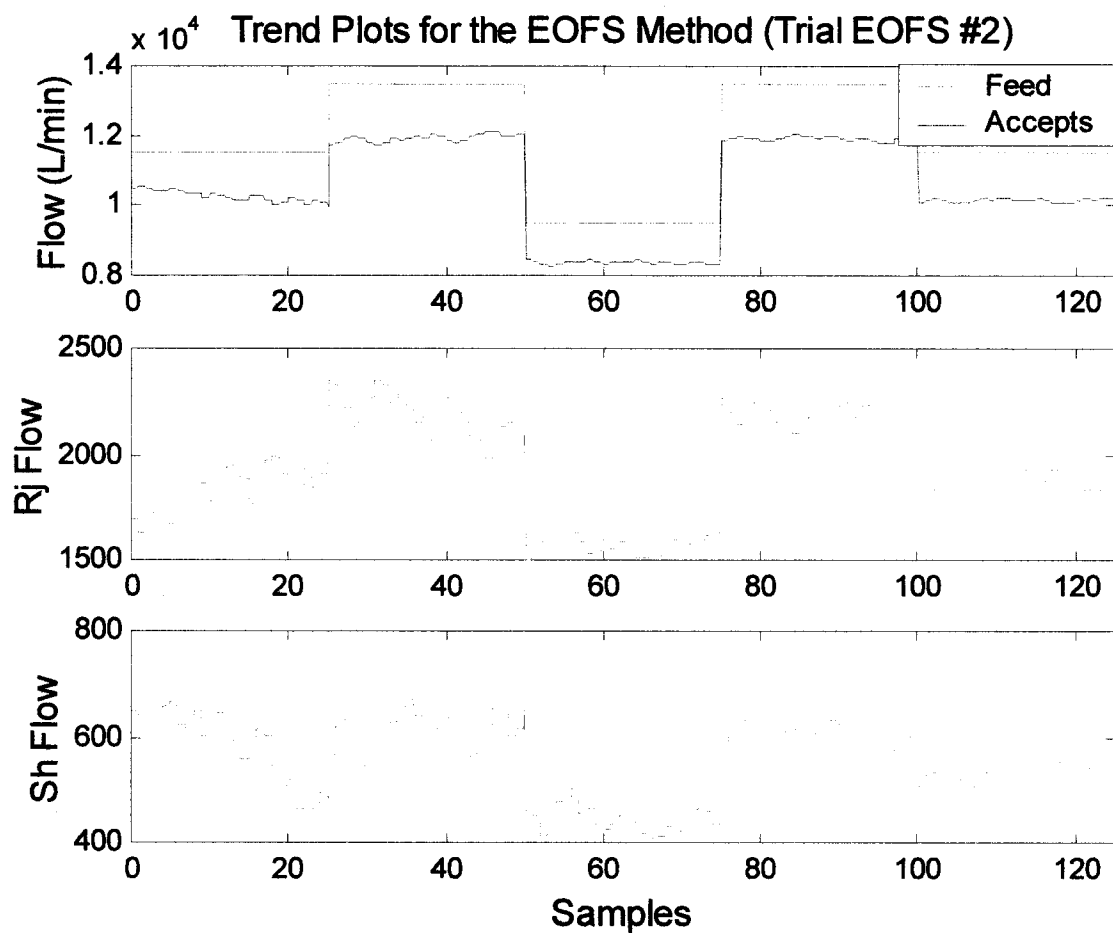


Figure 3.17(a): Trend plots for the flows of each stream for the simulation Trial EOFS #2.



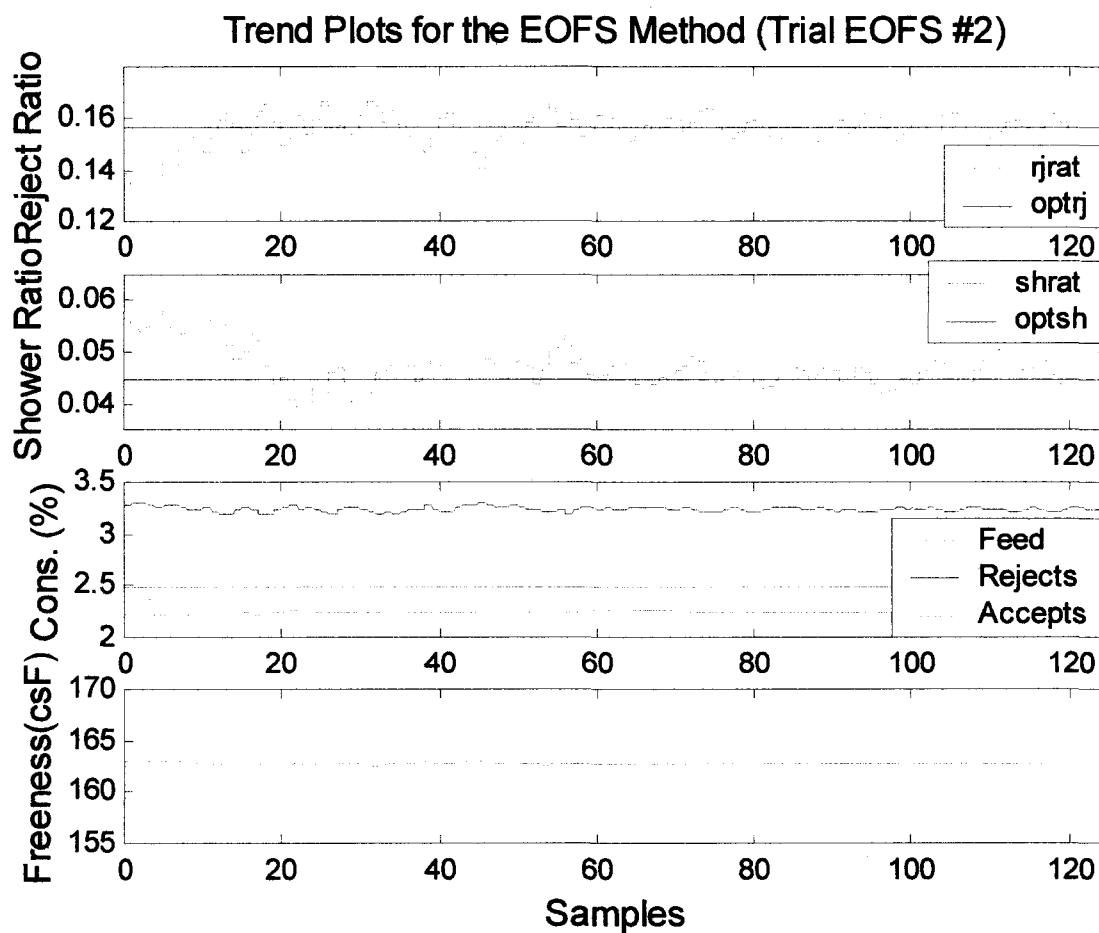


Figure 3.17(b): Trend plots for the operating variables for the simulation Trial EOFS #2.

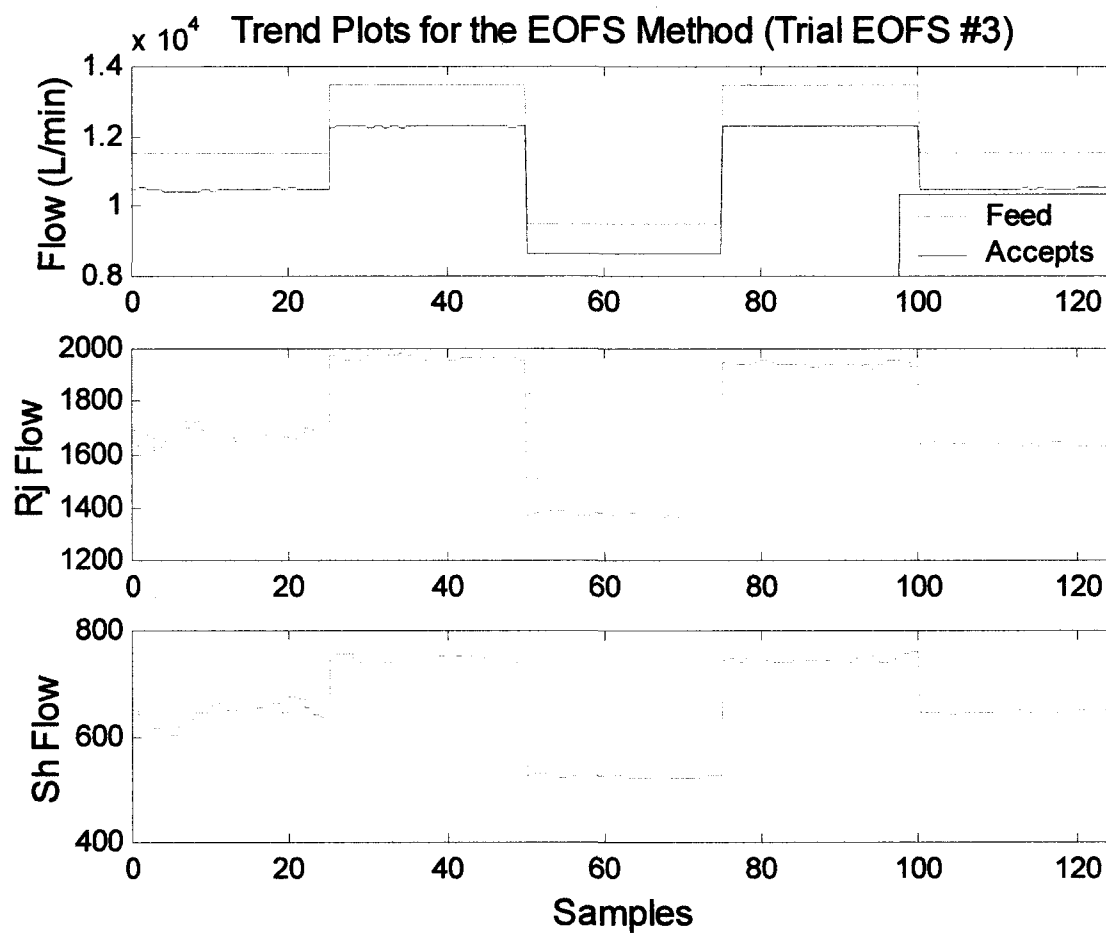


Figure 3.18(a): Trend plots for the flows of each stream for the simulation EOFS #3.

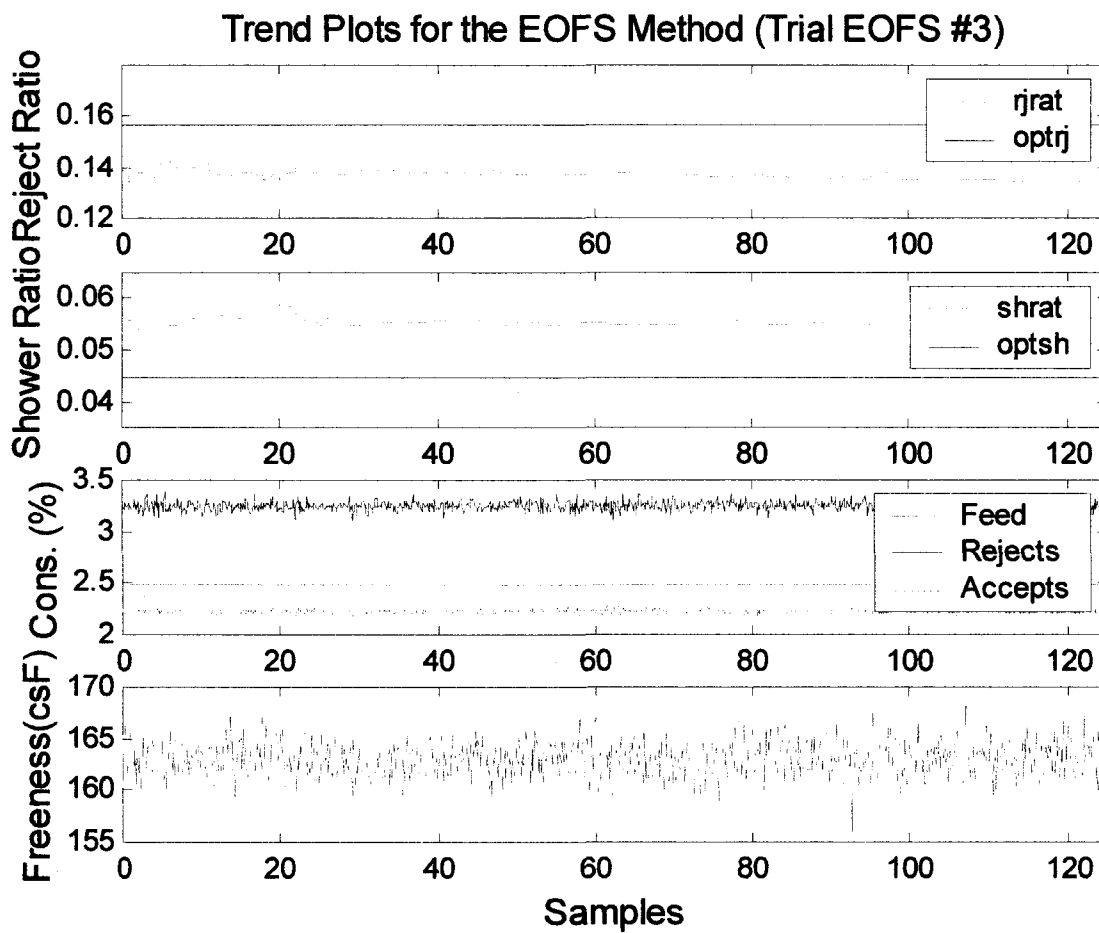


Figure 3.18(b): Trend plots for the operating variables for the simulation Trial EOFS #3.

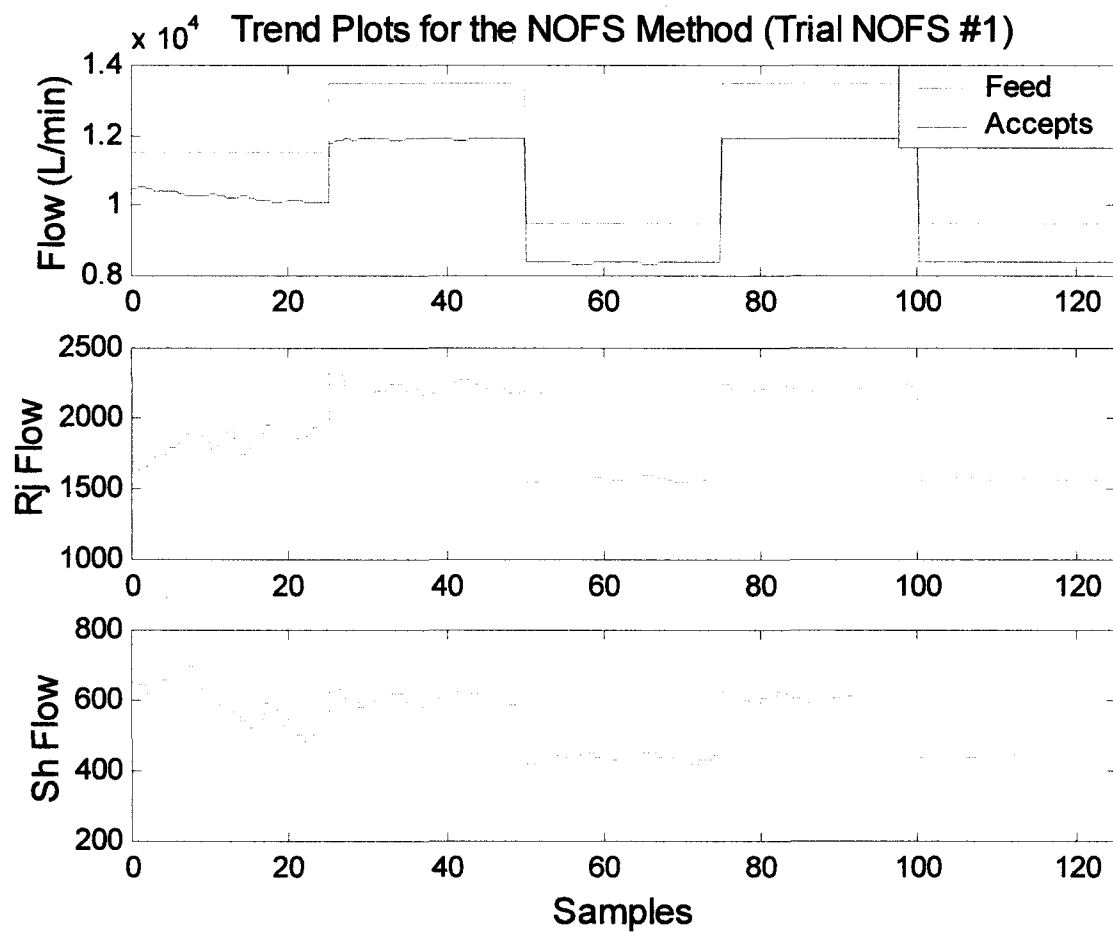


Figure 3.19(a): Trend plots for the flows of each stream for the simulation Trial NOFS #1.

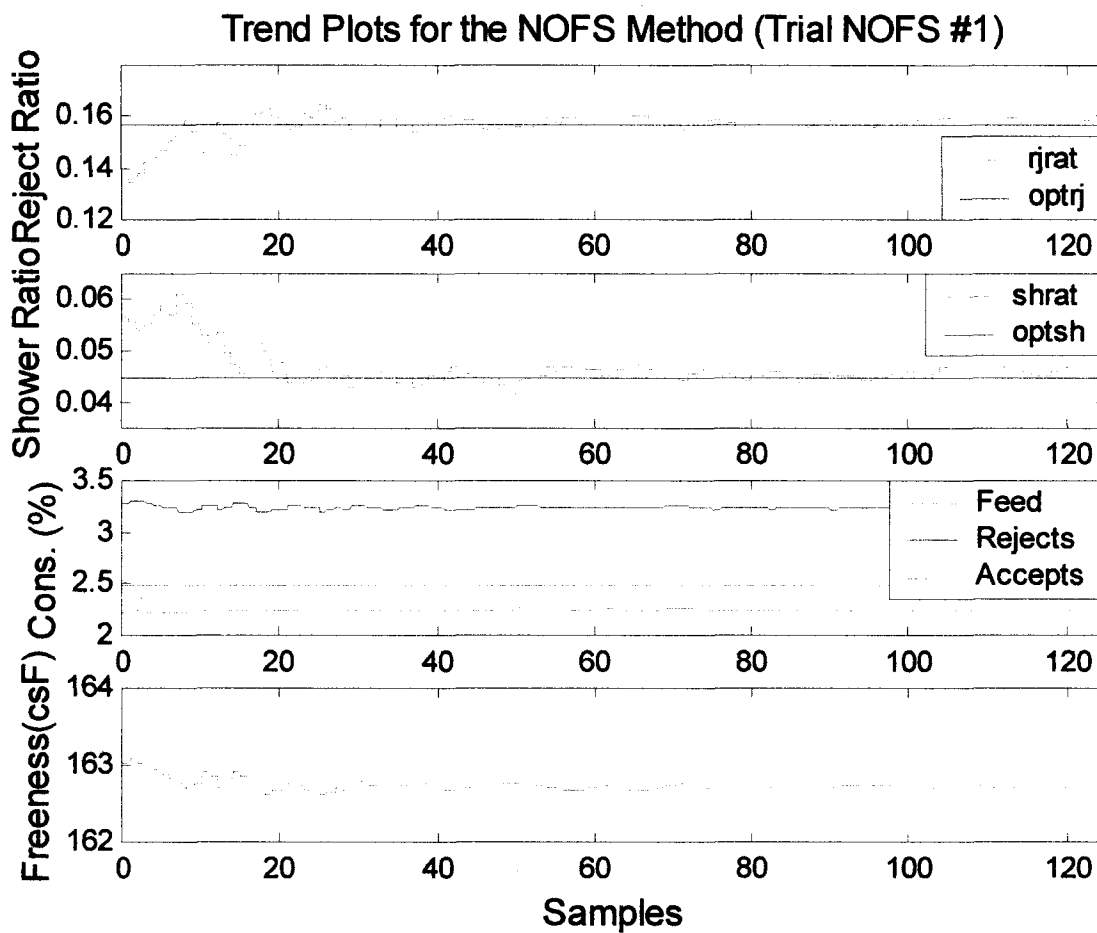


Figure 3.19(b): Trend plots for the operating variables for the simulation Trial NOFS #1.

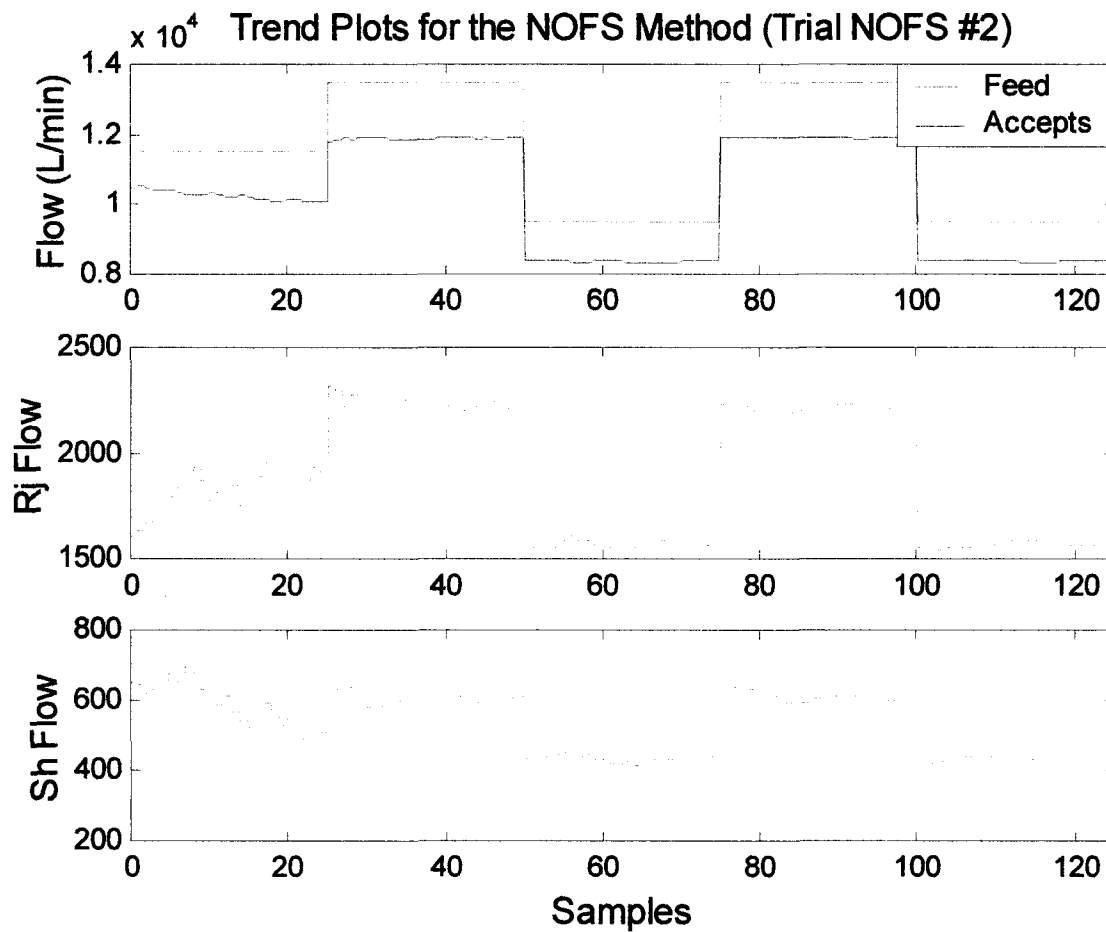


Figure 3.20(a): Trend plots for the flows of each stream for the simulation Trial NOFS #2.

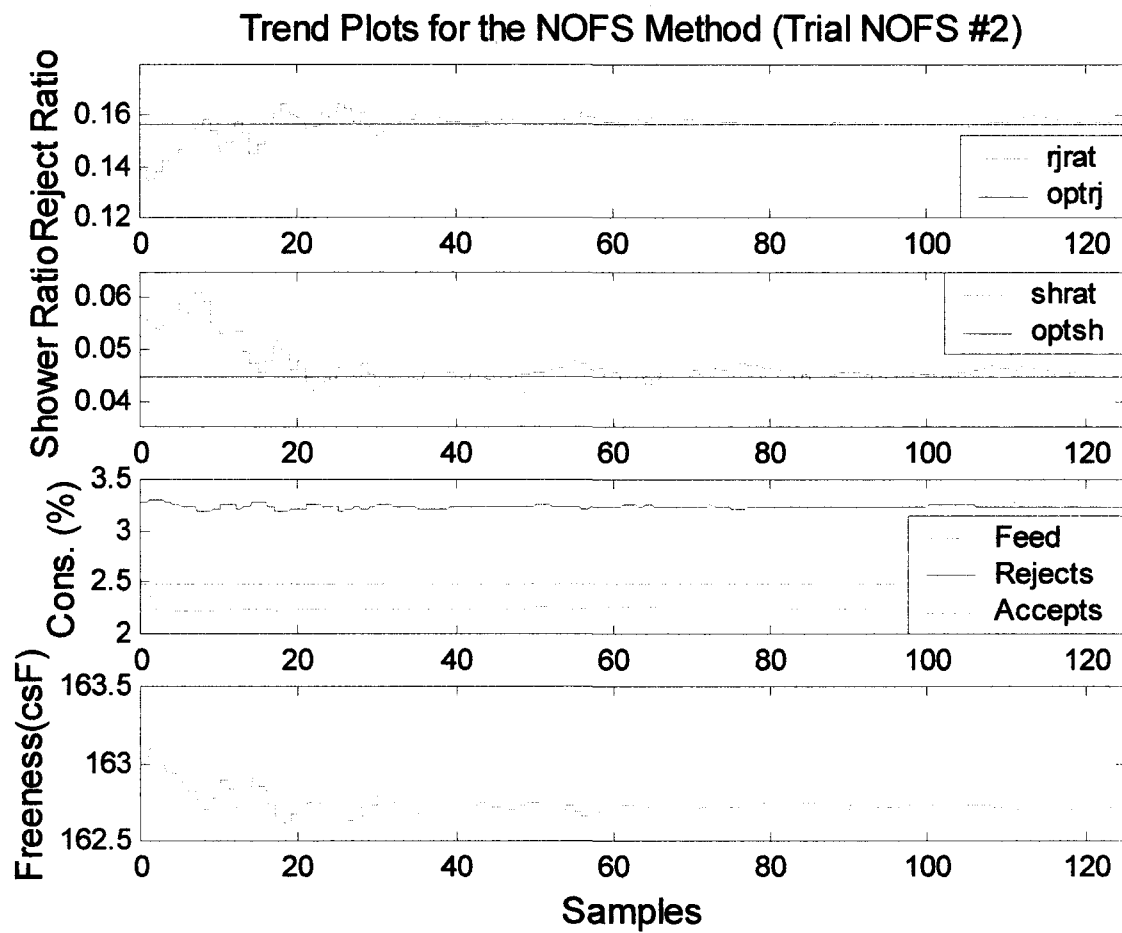


Figure 3.20(b): Trend plots for the operating variables for the simulation Trial NOFS #2.

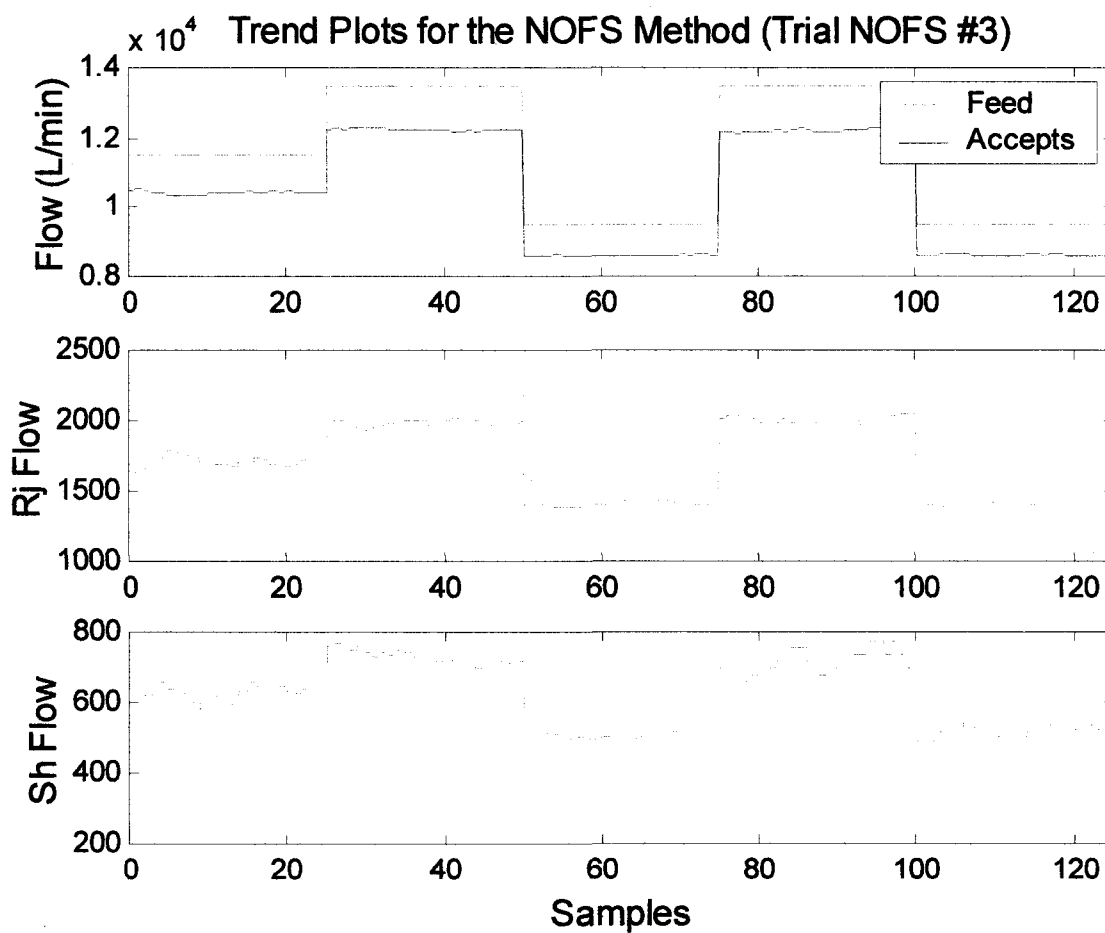


Figure 3.21(a): Trend plots for the flows of each stream for the simulation NOFS #3.



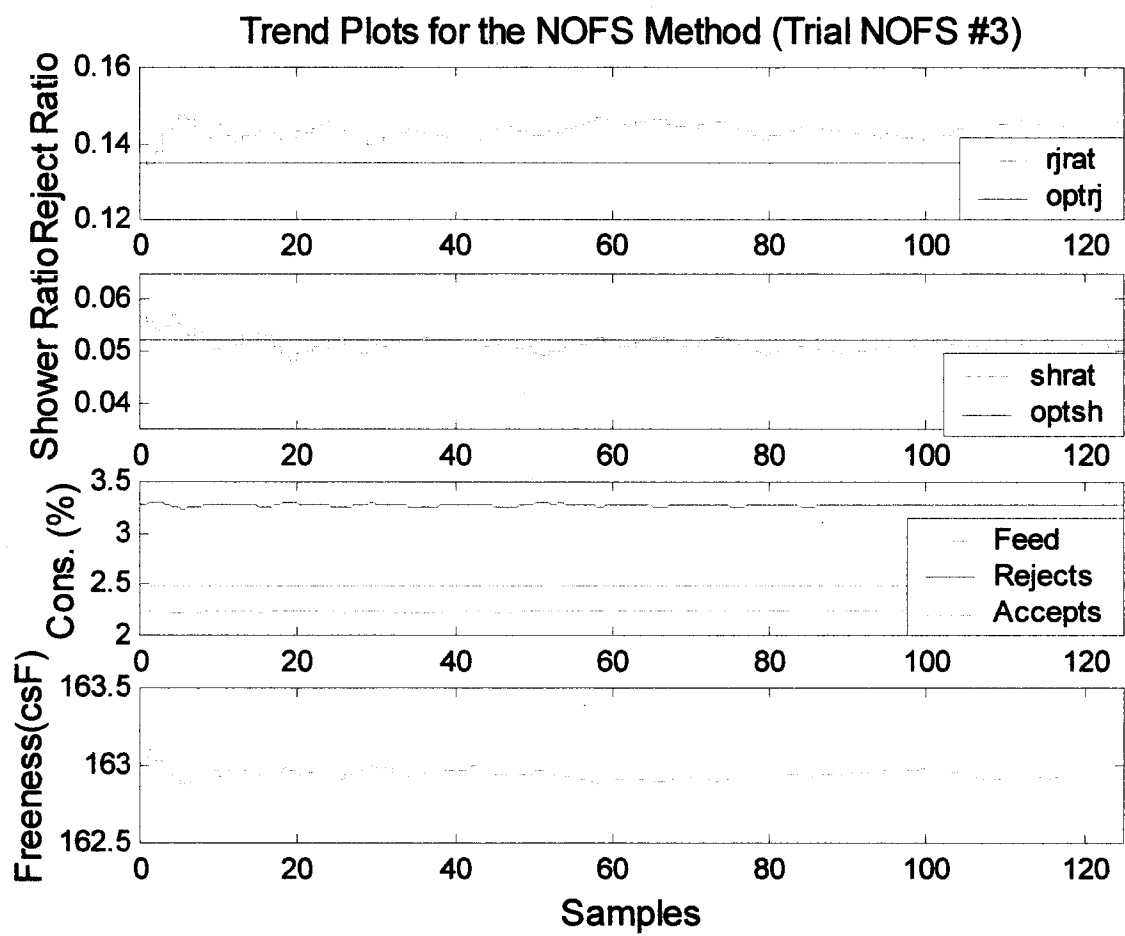


Figure 3.21(b): Trend plots for the operating variables for the simulation Trial NOFS #3.

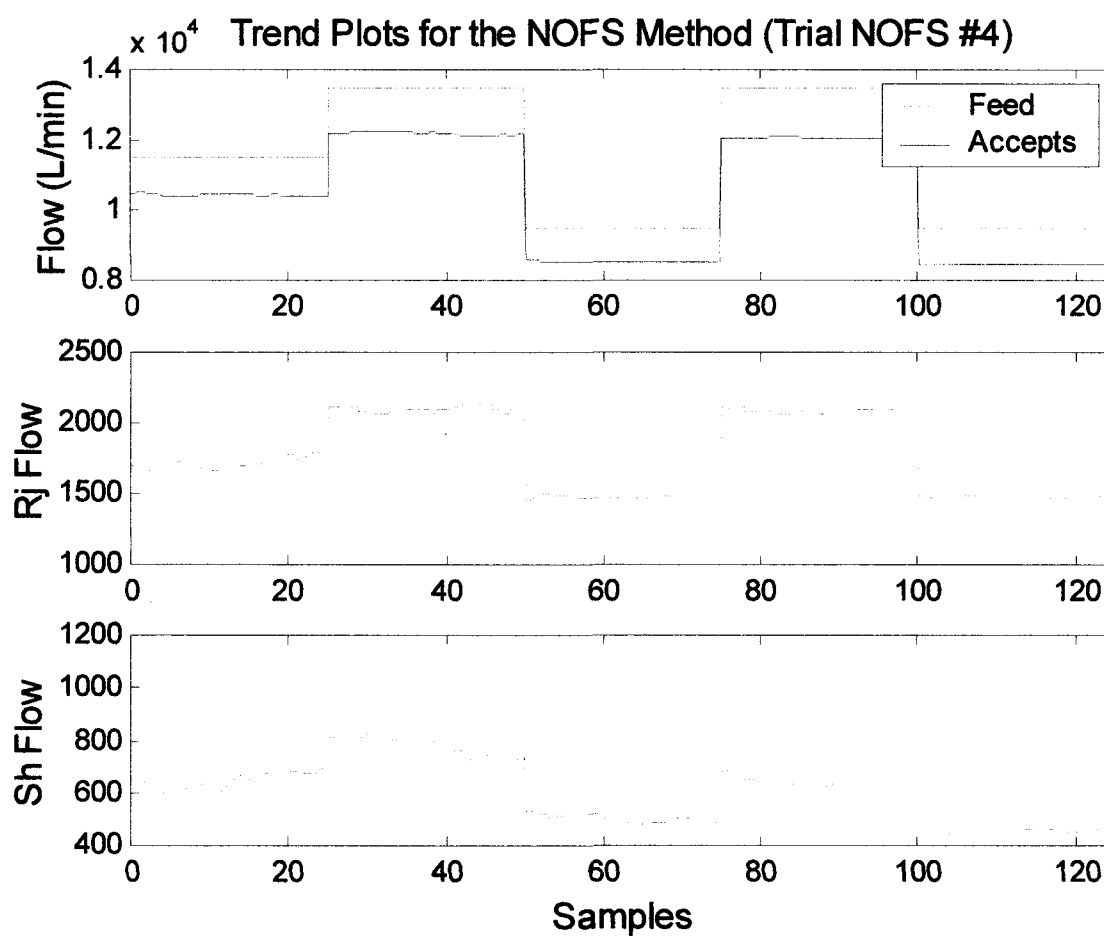


Figure 3.22(a): Trend plots for the flows of each stream for the simulation Trial NOFS #4.

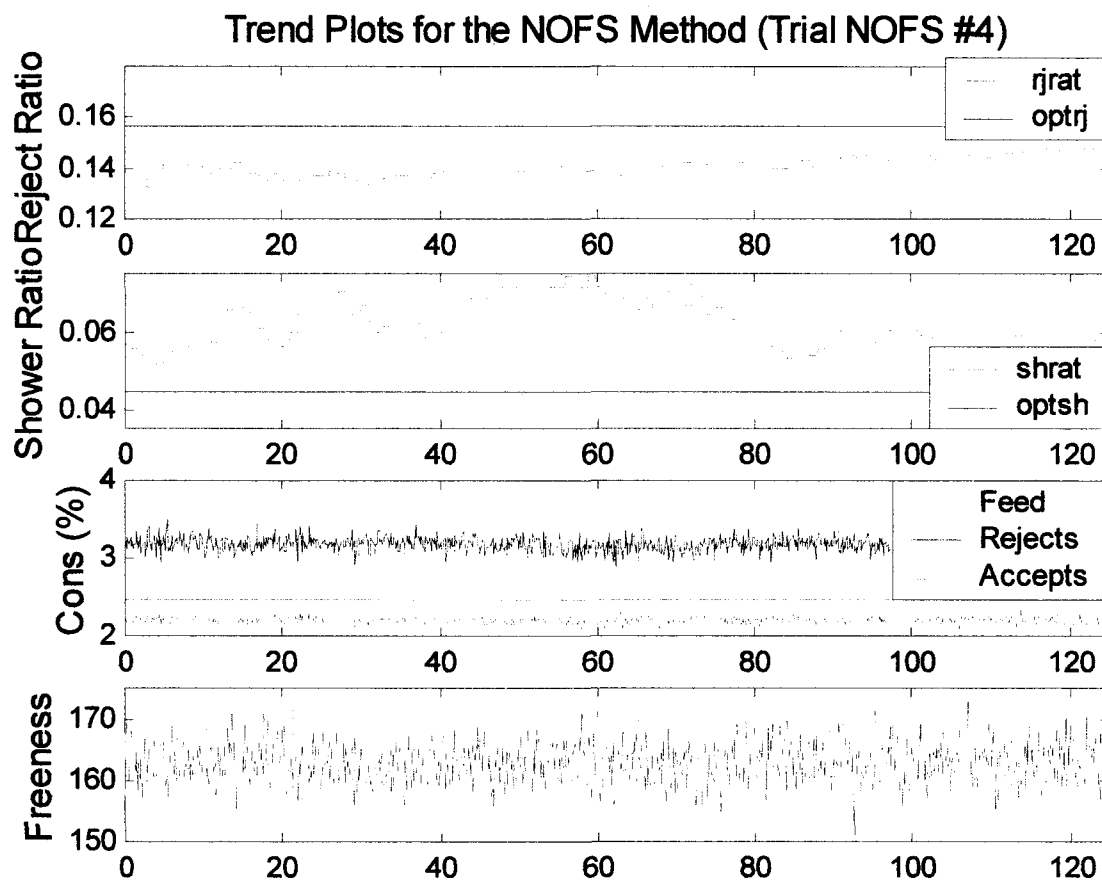


Figure 3.22(b): Trend plots for the operating variables for the simulation Trial NOFS #4.

A change in the feed fibre distribution is the second type of disturbance tested against the each of the simplex methods. The simulation of the pressure screen during a period in which the feed fibre distribution is altered includes some estimated results. As discussed in Chapter 1, modelling of the accepts freeness is an estimated equation based on samples taken from the plant. As a result of using this estimation, the weighting placed on the freeness term in the cost equation is reduced. Lowering the weighting factor on the cost of the freeness reduces the response of the simplex to changes in freeness as it responds to other disturbances. The ability of either simplex method to handle disturbances to the feed fibre distribution would be better in an industrial setting where the changes in freeness are measured, and not modelled using an estimated equation.

Figures 3.23 through 3.29 illustrate the simulations that are based on the feed fibre distribution disturbance. The disturbance to the feed fibre distribution was accomplished by using equation 3.30 and adjusting the value of  $\mu$ . The value of  $\mu$  was adjusted from 1.35 to 1.30 at the 50<sup>th</sup> sample, and increased back to 1.35 at the 88<sup>th</sup> sample.

$$\text{Feed Probability Distribution} = \frac{1}{l\sqrt{2\pi\beta}} e^{\frac{-(\ln l - \alpha)^2}{2\beta^2}} \quad (3.30)$$

Where 
$$\alpha = \ln \mu - \frac{\beta^2}{2}$$

And 
$$\mu = 1.35 \text{ and } \beta = 1.12$$

Only one of the simulations without noise is shown. The simulation using the expansion factor of 2.0 and the contraction factor of 0.5 was omitted because the other factors were shown to be better in the previous simulations. Some of the operating variables for the simulations shown in Figures 3.23 and 3.24 that use the EOFS method are as follows.

- starting simplex size = 0.005
- minimum simplex size = 0.000625
- maximum simplex size = 0.011

Figures 3.23(a) and (b) illustrate Trial EOFS #4. The expansion was 1.25 and the contraction factor was 0.8 for this simulation. The results of the simulation with respect to optimum tracking was excellent. The optimum shower ratio was unchanged but the optimum reject ratio shifted when the disturbance entered the

simulation. The simplex followed the moving optimum very well, but no correction to the freeness is noticeable. The lack of correction is probably due to a combination of size of the disturbance and the inability of the screen model to make significant changes to the percentage of accepts fines.

Figures 3.24(a) and (b) illustrate Trial EOFS #5, which tests the ability of the EOFS method to handle a feed fibre distribution disturbance. Because of the noise in the simulation it is difficult to notice if the EOFS method is able to correct the accepts freeness. The size of the oscillations are very small. The minimum size of the simplex should have been larger to better respond to change. Although the shower ratio nearly matches its optimum, the reject ratio doesn't respond to the change in its optimum when the disturbance occurs. This may be a result of the noise added to the simulation, or because the weighting of the freeness term in the cost equation was reduced. Since the simplex responded so well in the previous simulation, the noise is most likely the cause of the problems in this simulation.

Some of the operating variables used in the simulations shown in Figures 3.25 through 3.29 are as follows:

- starting simplex size = 0.005
- minimum simplex size = 0.001
- maximum simplex size = 0.045

Figures 3.25(a) and (b) illustrate Trial NOFS #5, which uses no shape constraints. The minimum and maximum simplex size are still in place, it's the limits on how elongated the simplex can become that is removed in this simulation. Both of the ratio's settle down before the disturbance enters the simulation. Once the disturbance enters, the reject ratio begins to respond within 10 samples by decreasing. The decrease in the reject ratio reduces the flow through the reject valve enough to see an increase in the rejects consistency and a small decrease in the fines fraction of the accepts, which causes a small increase in the accepts freeness.

After the 88<sup>th</sup> sample, the incoming accepts freeness returned to its original value. At this time, both the reject and shower ratio's were oscillating back to their value prior to the disturbance at the 50<sup>th</sup> sample.

Figure 3.26(a) and (b) illustrate Trial NOFS #6, which uses the shape constraint. Similar to the previous simulation the reject ratio responded well to the disturbance that caused the accepts freeness to

increase at the 50<sup>th</sup> sample. In this trial the simplex appears to have a slower response than in the previous simulation.

Figures 3.27(a) and (b) illustrate Trial NOFS #7, which uses the shape constraint and replaces the maximum separation term of the cost equation with the target fibre distribution term. Although the simplex responded well to change in the optimum reject ratio, it did not perform as well as Trials NOFS #5 and #6. It may be that the weighting of the target fibre distribution term in the cost equation was not large enough. Otherwise, the shower ratio responded as expected in an attempt to keep the rejects consistency within its acceptable range. Some of the operating variables used in the NOFS #7 simulation are as follows:

- starting simplex size = 0.005
- minimum simplex size = 0.002
- maximum simplex size = 0.005

Figures 3.28(a) and (b) illustrate Trial NOFS #8, which uses the shape constraint along with the addition of noise to simulate actual plant conditions. This simulation is the same as the one in Figures 3.26(a) and (b), but with the addition of the noise. In this simulation the noise was removed from the accepts fine value. When the noise was added to the process it prevented the disturbance from being recognized. Since the fraction of accepts fine is not part of the cost equation for this simulation, the noise was removed from that term. The response to the disturbance in this simulation is different than what is seen in Figures 3.26(a) and (b). There is very little response from either the reject or shower ratio to the disturbance. In previous simulations the reject ratio decreased when the disturbance occurred. The shower ratio responded with a steady increase throughout the disturbance period. As with other simulations that include noise, the simplex does not work very well.

Xiong and Jutan [19] discovered that in the presence of noise their simplex performed better when the location of the measurements were well spaced. A problem with well spaced measurements occurs in a process like optimizing a pressure screen. At steady state, it is expected that each measurement will be similar, and the simplex will shrink to the optimum. In order to have well spaced measurements, you must have a large simplex where each vertex is well spaced.

To determine the problem, the simulations with noise added were run using the same code with the noise set to zero. A single line of code was added to set the value of the noise to zero. This result is

shown in Figures 3.29(a) and (b) and is labelled as Trial NOFS #8a. The results of Trial NOFS #8a indicate that the failed simulations are a result of the noise. The simplex will work as long as noise isn't present in the simulation. If noise is the problem in the failed simulations it would seem reasonable that the reason behind the problem is the inconsistent measurement results. The simplex probably never gets any momentum in one direction. Every time the simplex starts to move in one direction a measurement will cause the simplex to move the opposite direction. If the simplex flips back and forth because of the noise it may never move away from its original location unless the optimum has moved a significant distance away. Solving this problem can be handled using a running average of the measurement analysis. This method would work great if the sampling period were shorter. With the current PQM setup at Bowater the sampling period is 40 minutes. If the running average was based on five samples, the running average would be based partially on a sample taken 160 minutes prior to the most current sample. Obviously this method would give meaningless results, and not give a response to any short term disturbance.

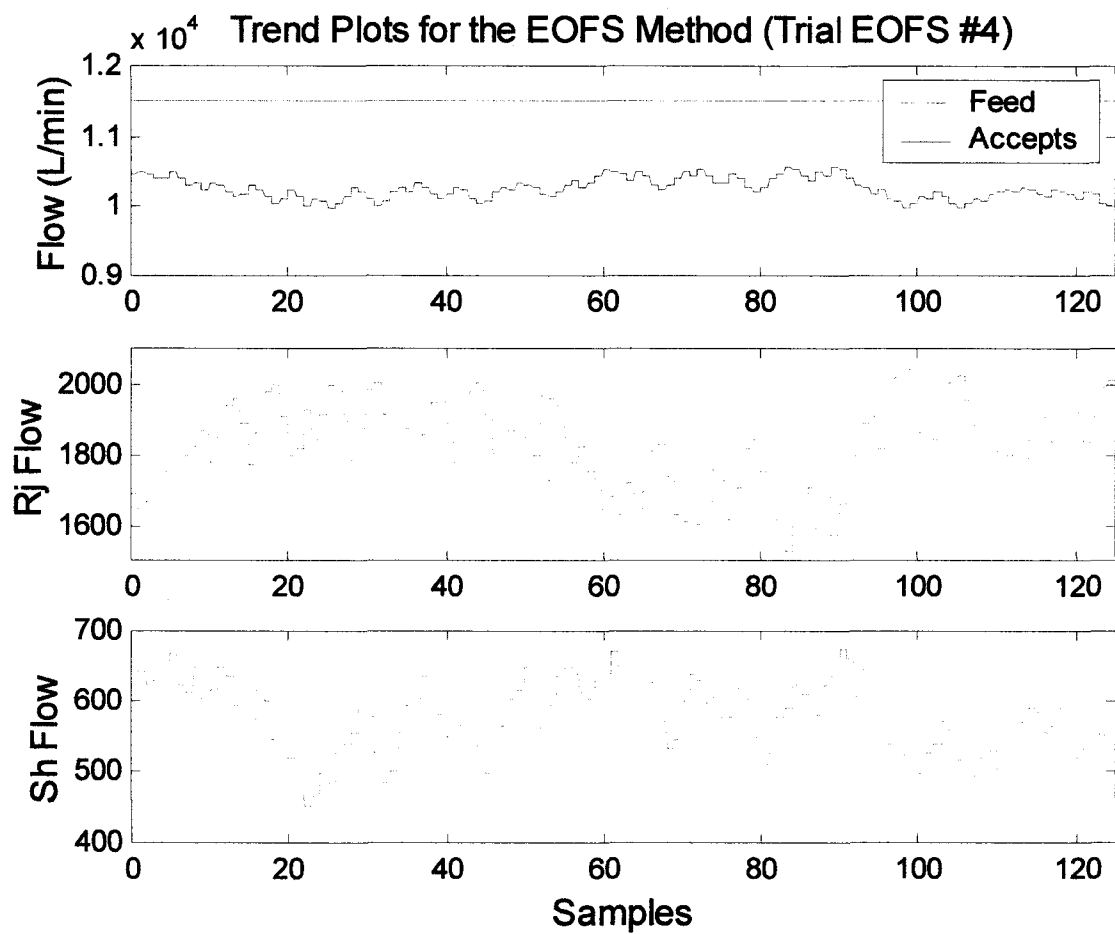


Figure 3.23(a): Trend plots for the flows of each stream for the simulation Trial EOFs #4.



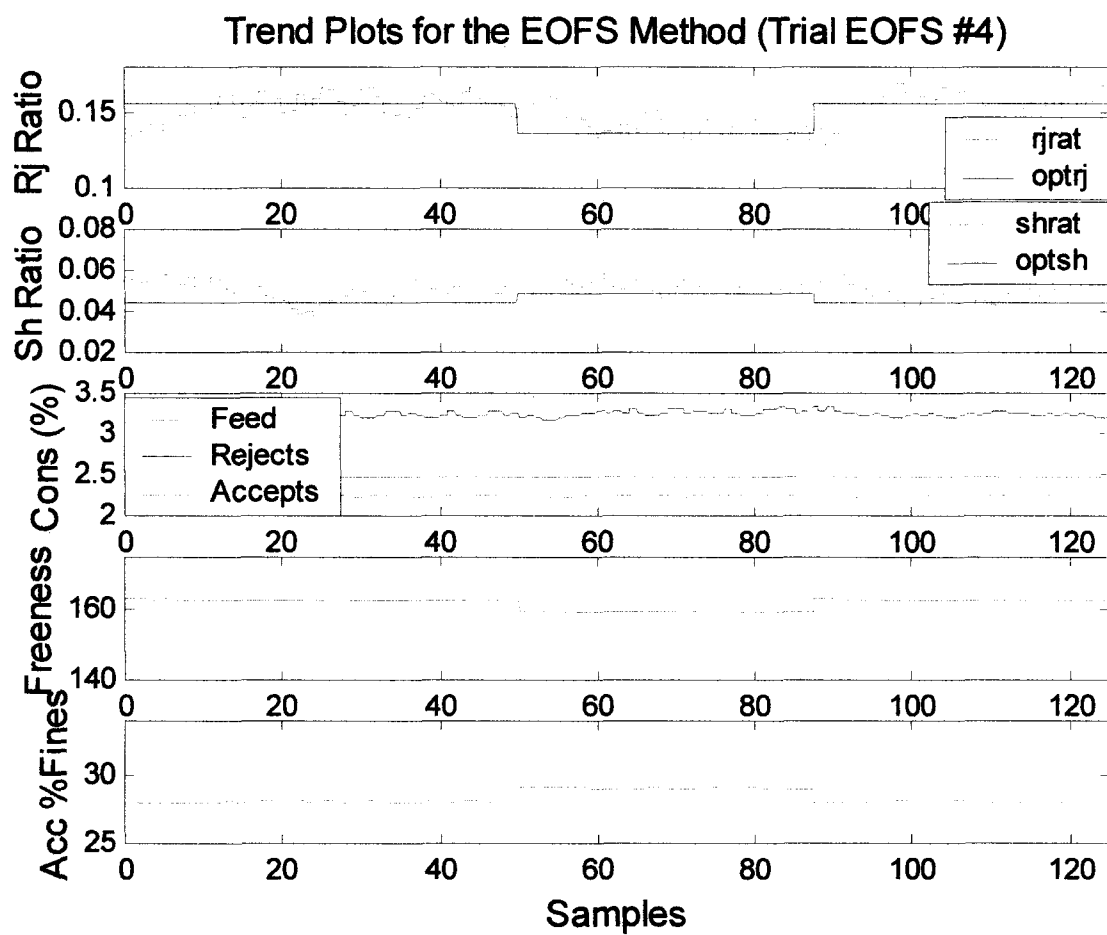


Figure 3.23(b): Trend plots for the flows of each stream for the simulation Trial EOFS #4.

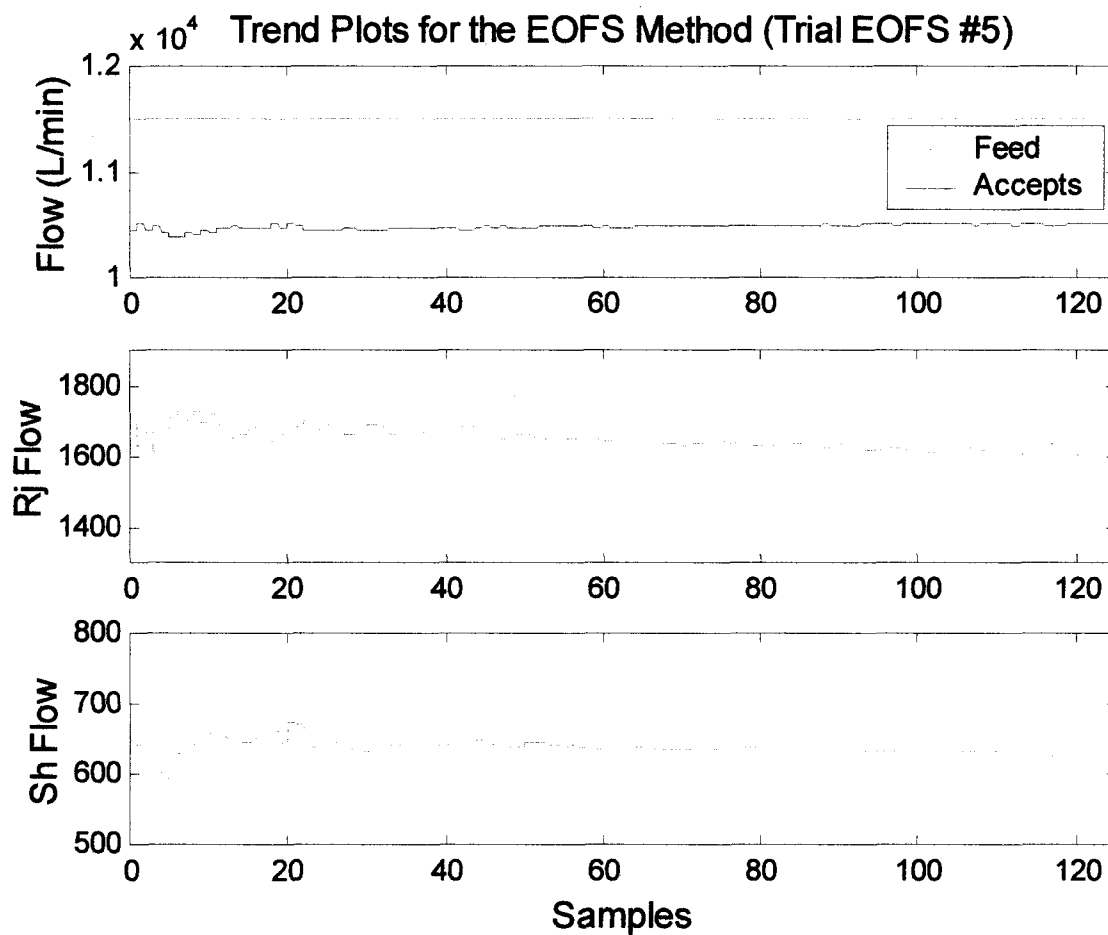


Figure 3.24(a): Trend plots for the operating variables for the simulation Trial EOFS #5.

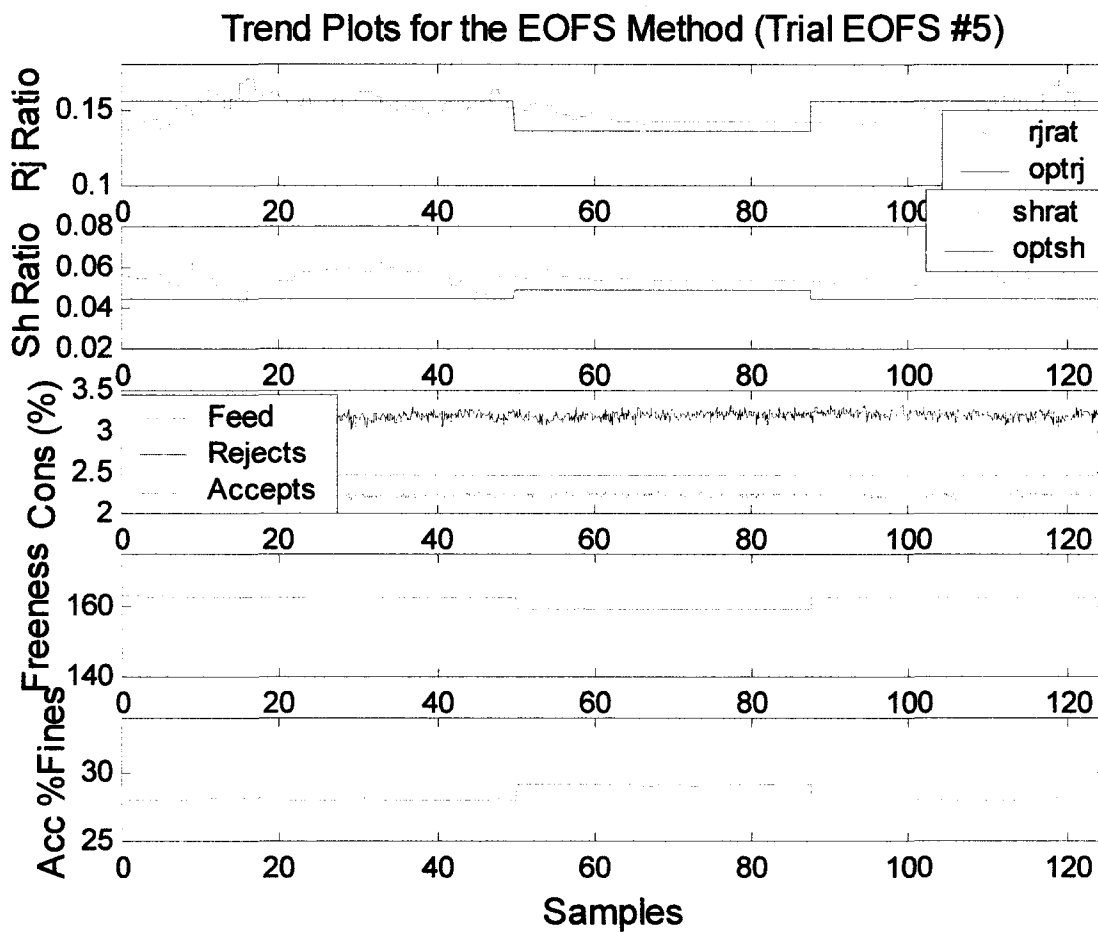


Figure 3.24(b): Trend plots for the operating variables for the simulation Trial EOFS #5.

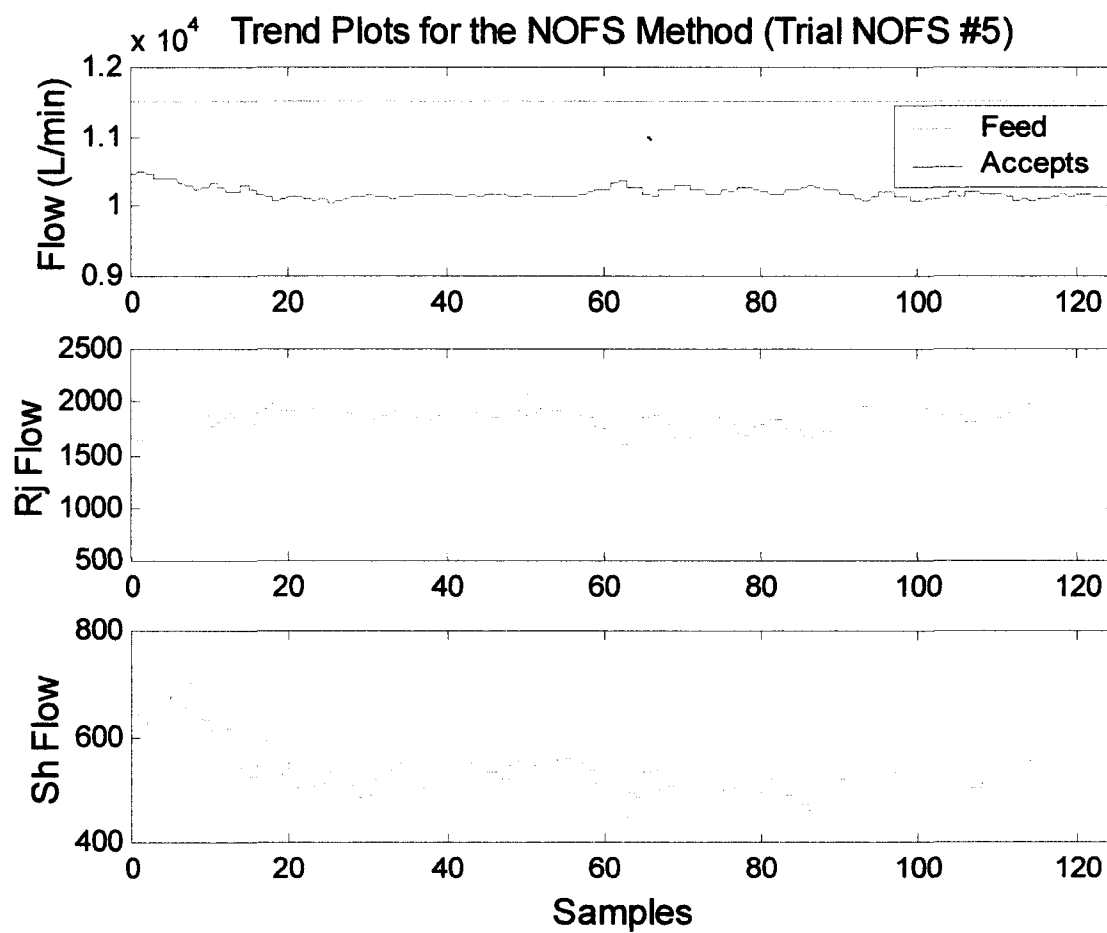


Figure 3.25(a): Trend plots for the flows of each stream for the simulation Trial NOFS #5.

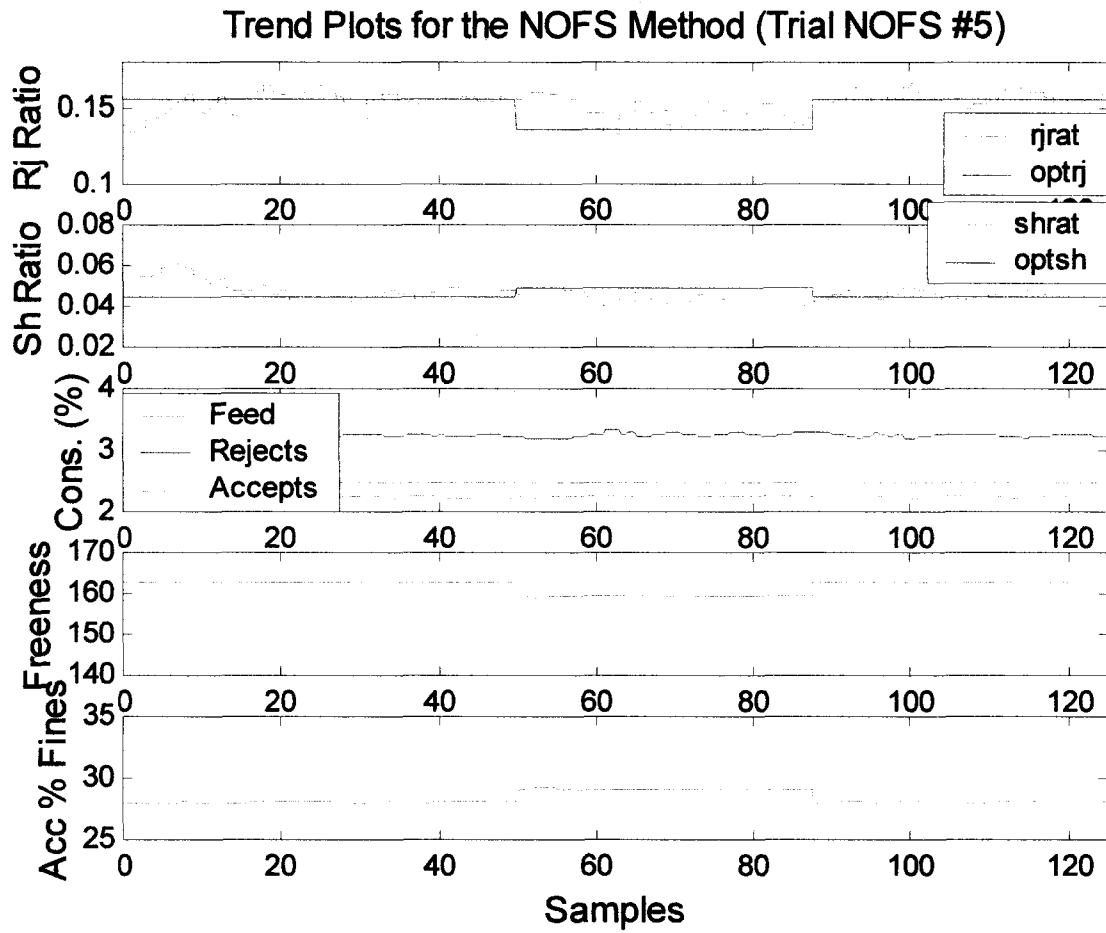


Figure 3.25(b): Trend plots for the operating variables for the simulation Trial NOFS #5.

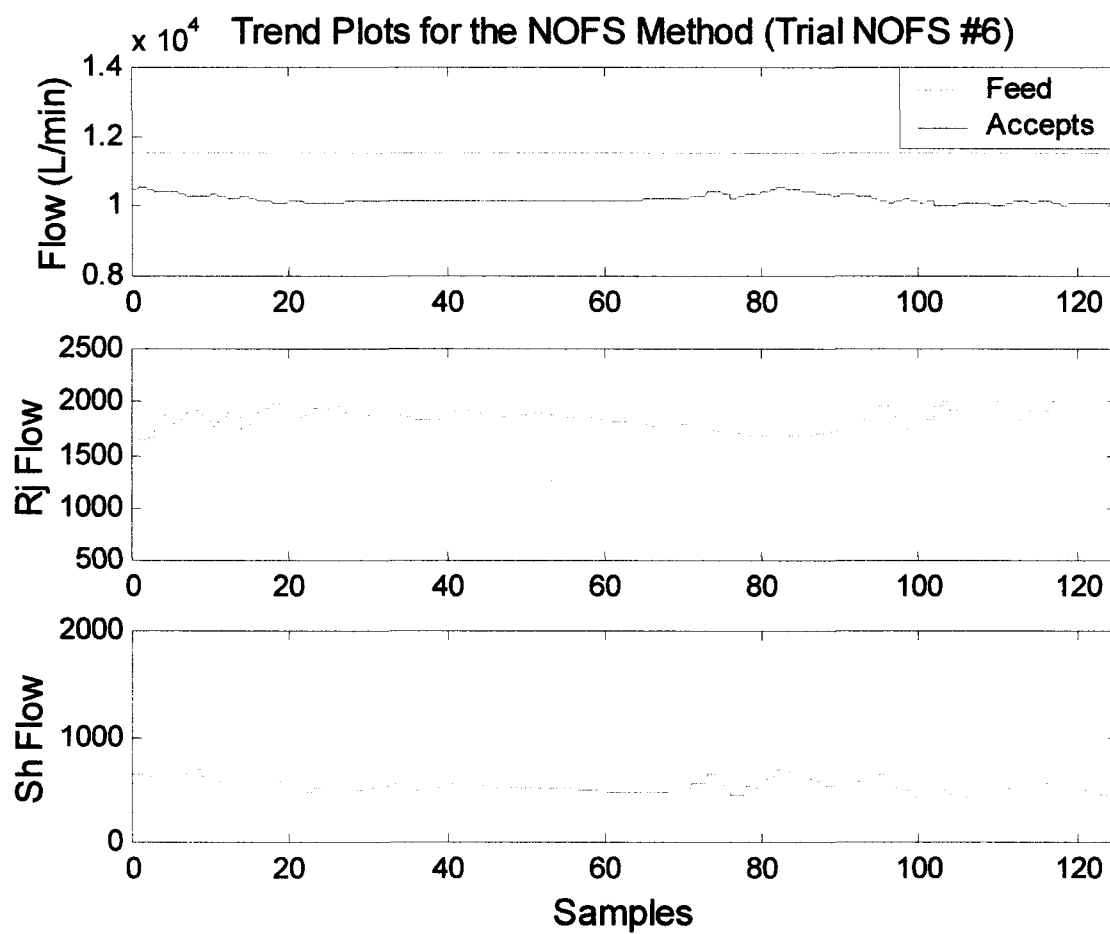


Figure 3.26(a): Trend plots for the flows of each stream for the simulation Trial NOFS #6.

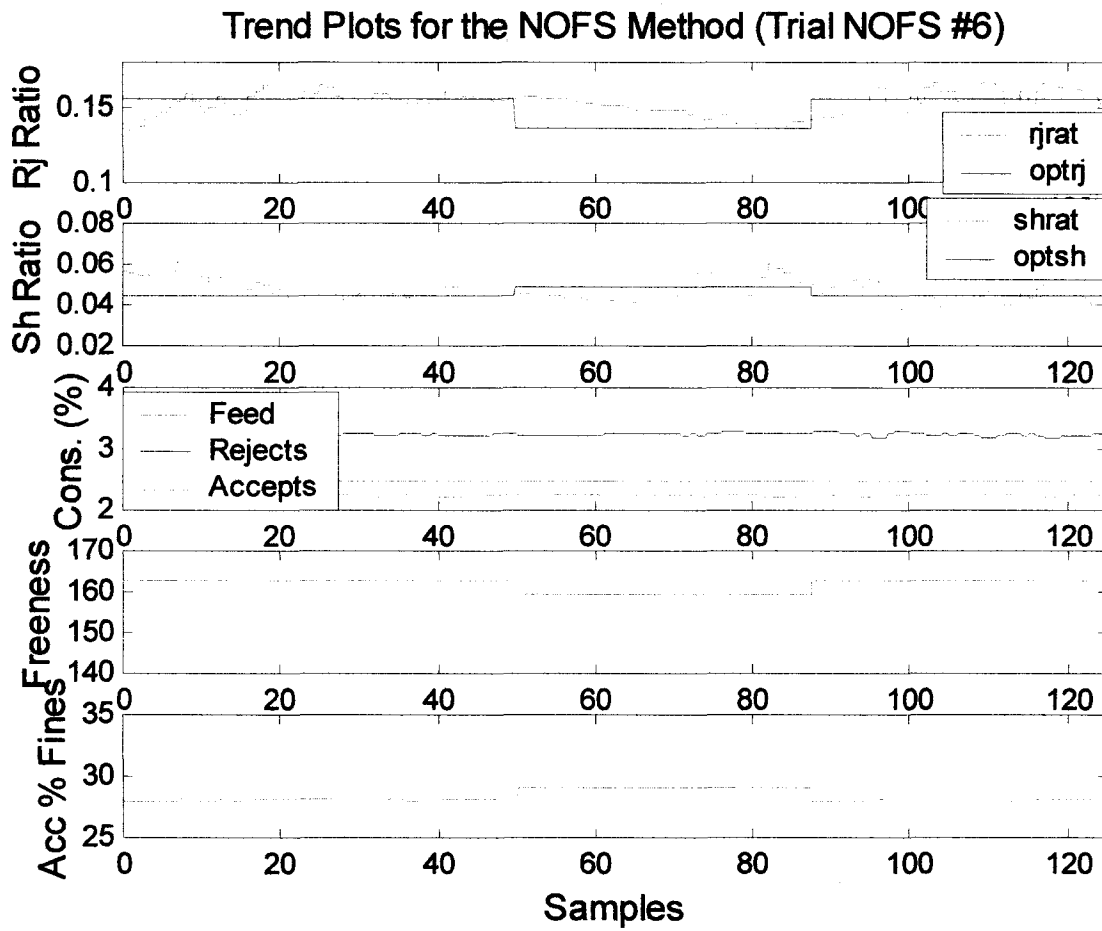


Figure 3.26(b): Trend plots for the operating variables for the simulation Trial NOFS #6.

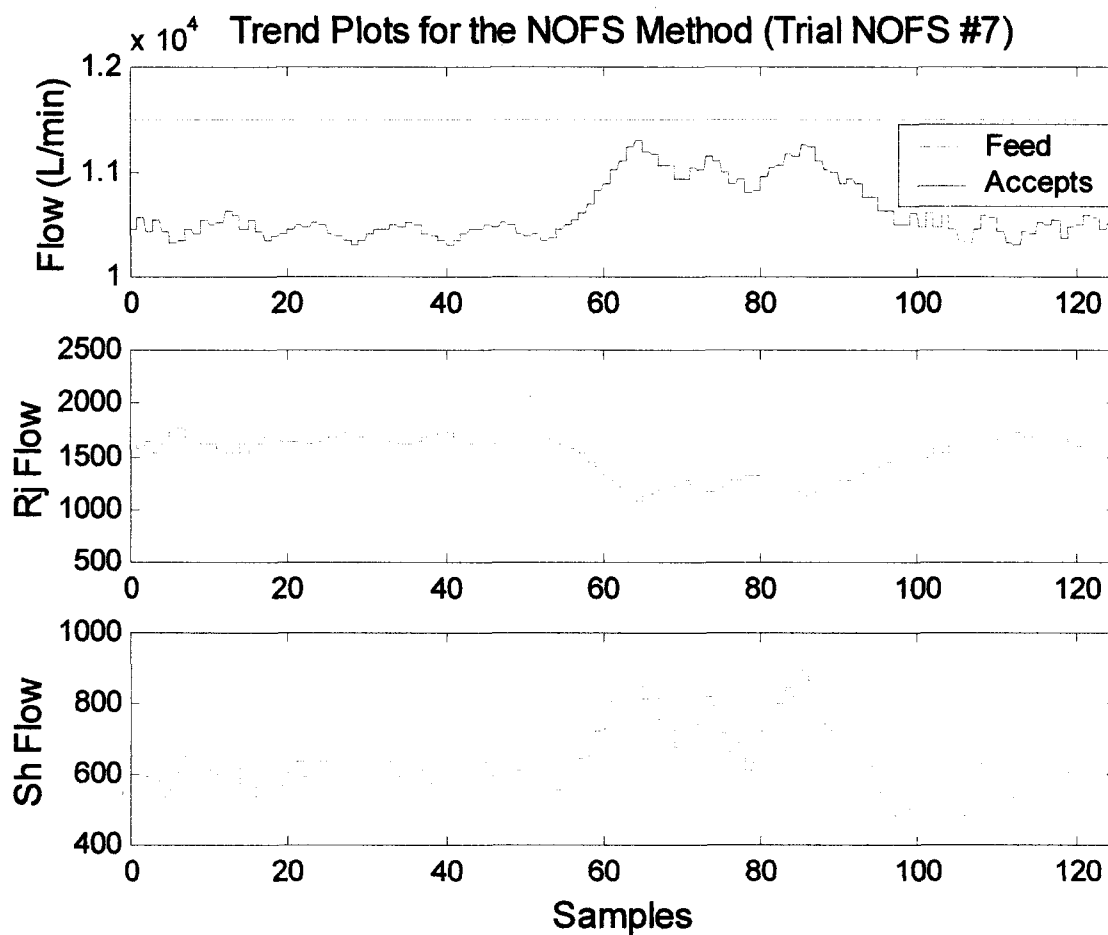


Figure 3.27(a): Trend plots for the flows of each stream for the simulation Trial NOFS #7.



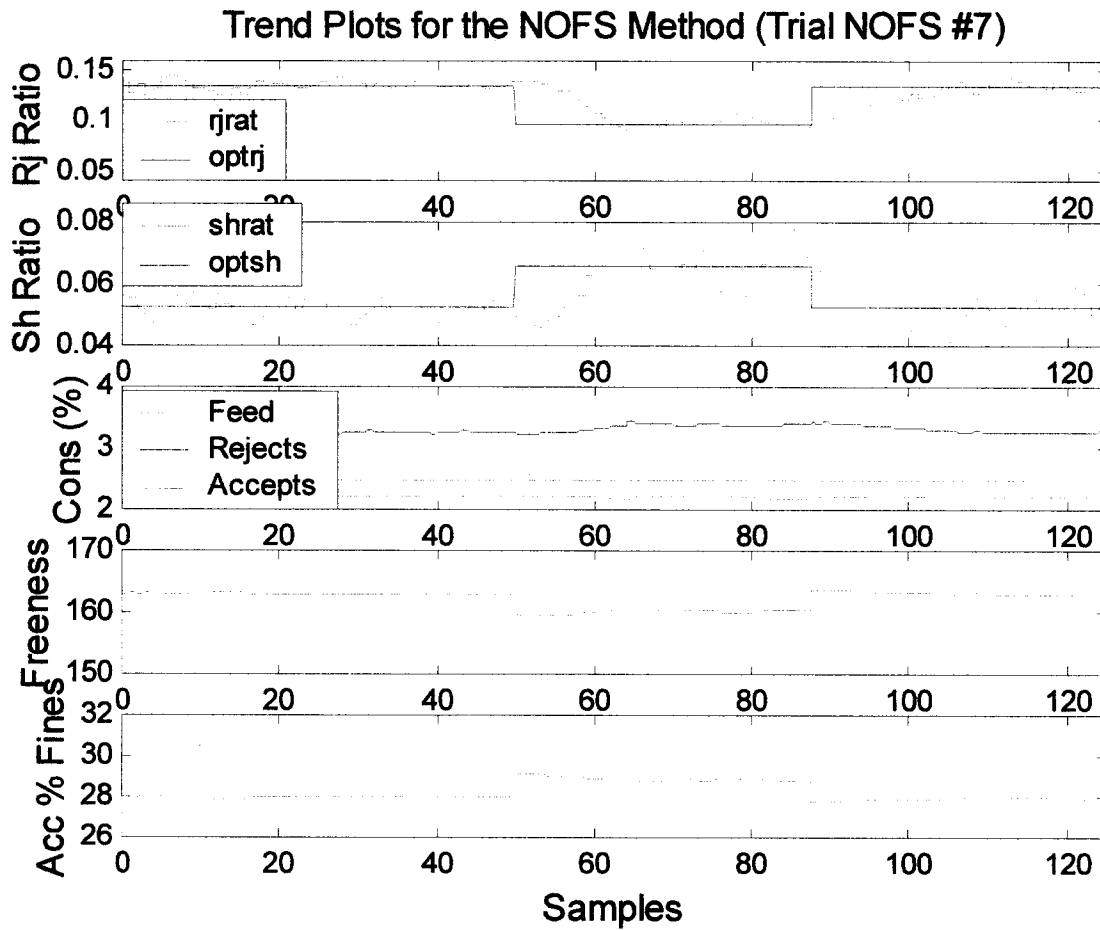


Figure 3.27(b): Trend plots for the operating variables for the simulation Trial NOFS #7.

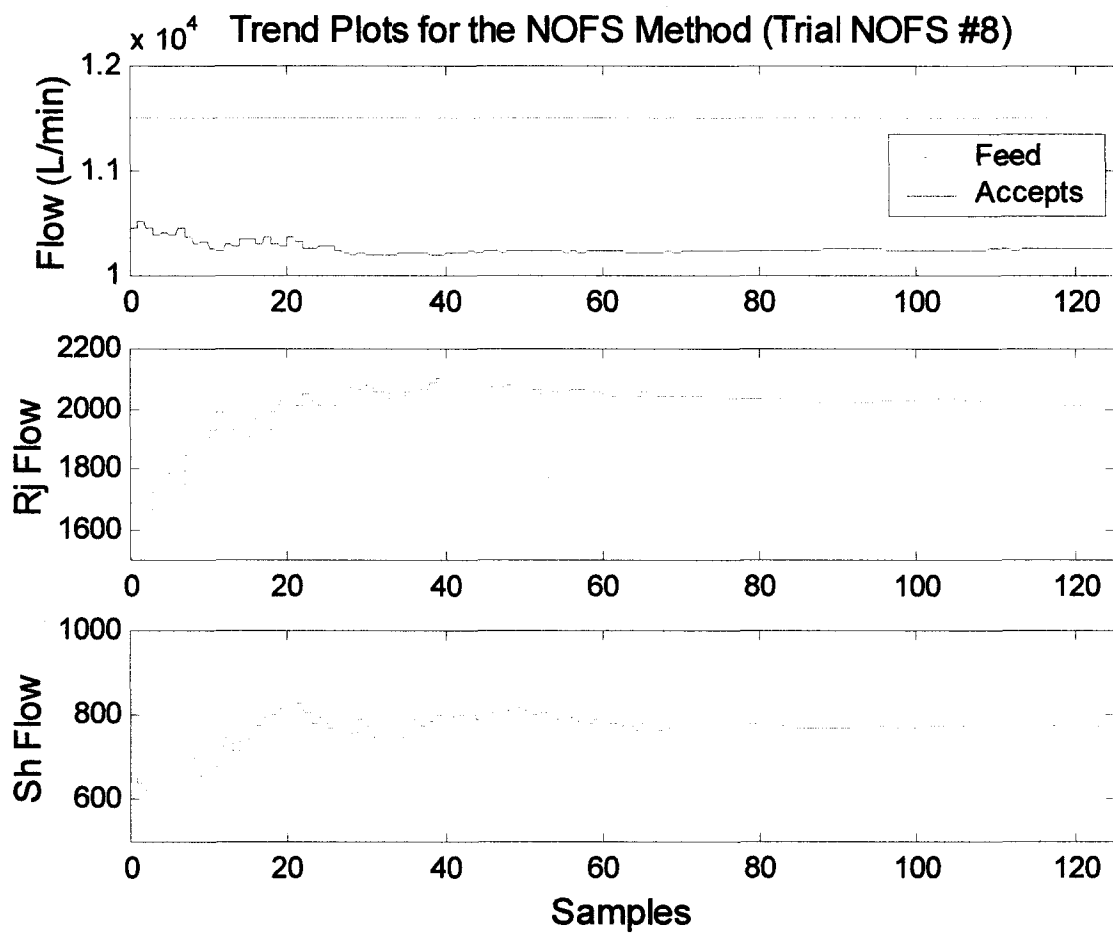


Figure 3.28(a): Trend plots for the flows of each stream for the simulation Trial NOFS #8.

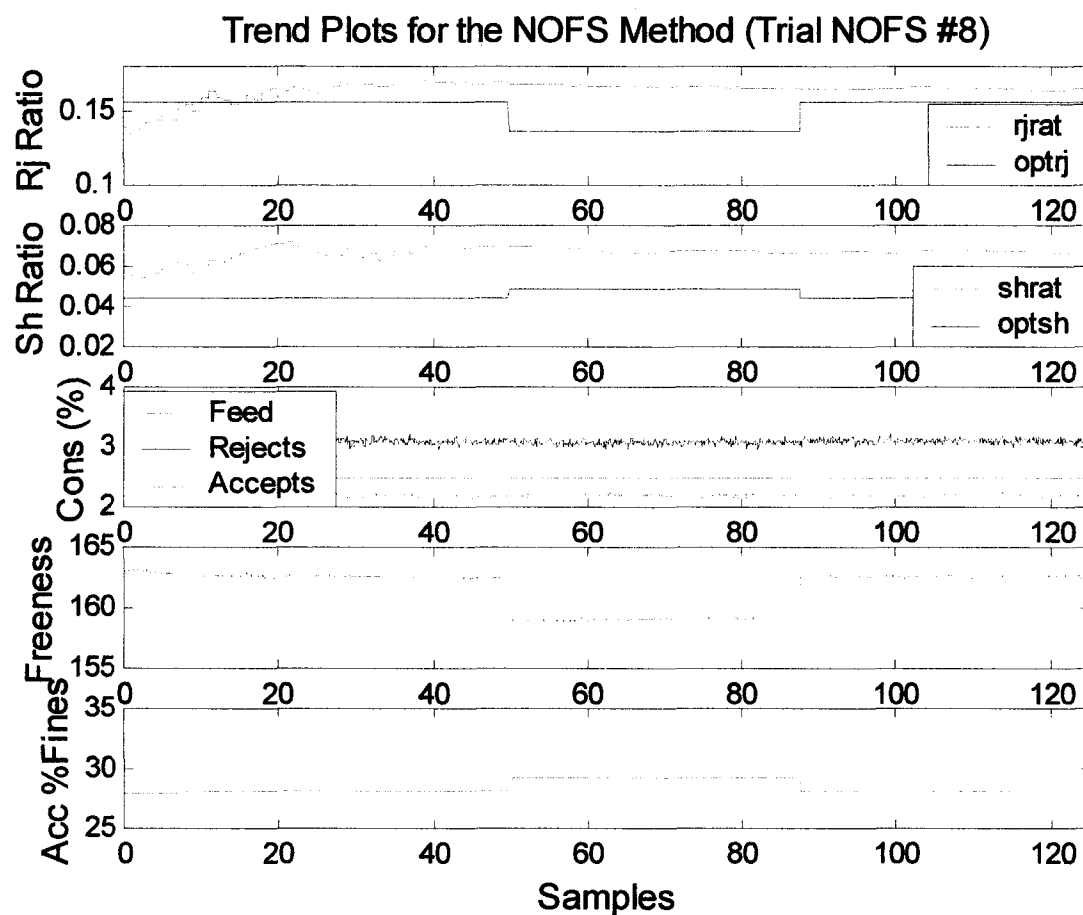


Figure 3.28(b): Trend plots for the operating variables for the simulation Trial NOFS #8.

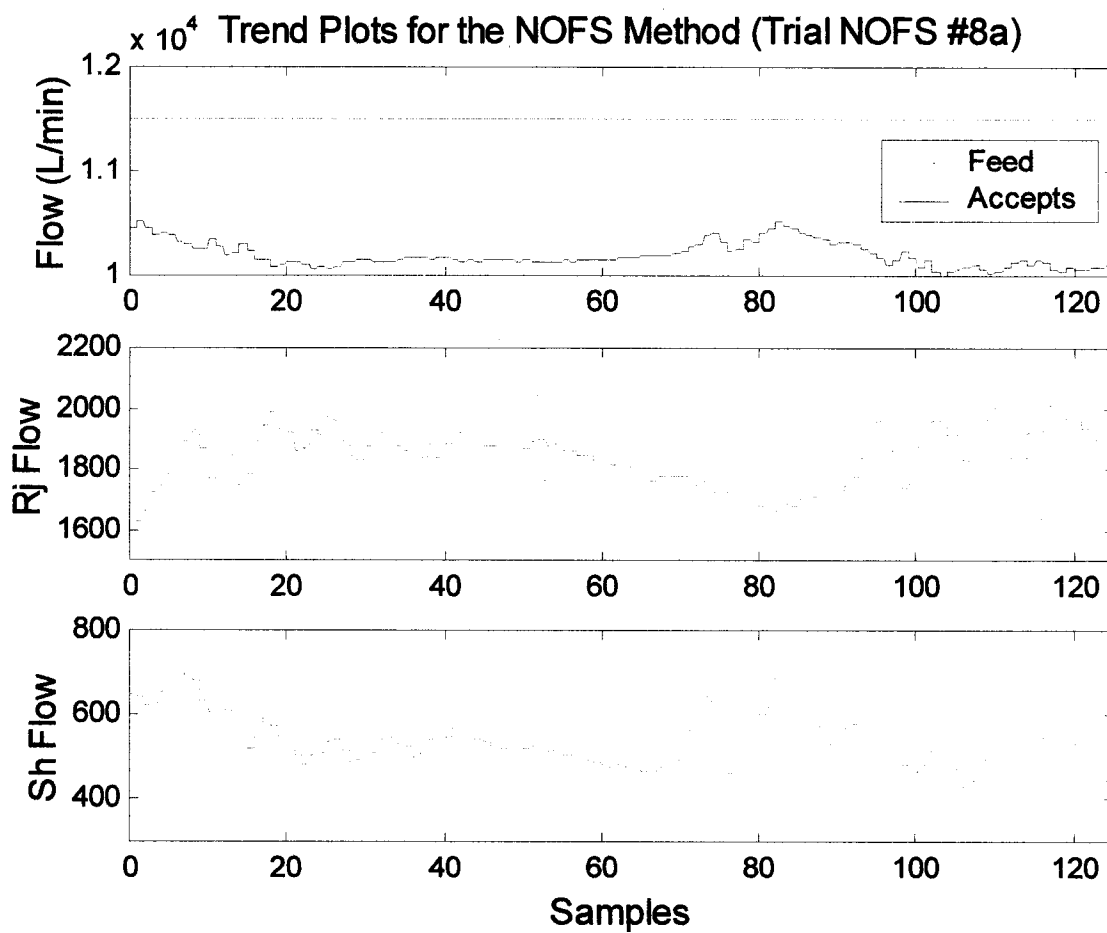


Figure 3.29(a): Trend plots for the operating variables for the simulation Trial NOFS #8a.

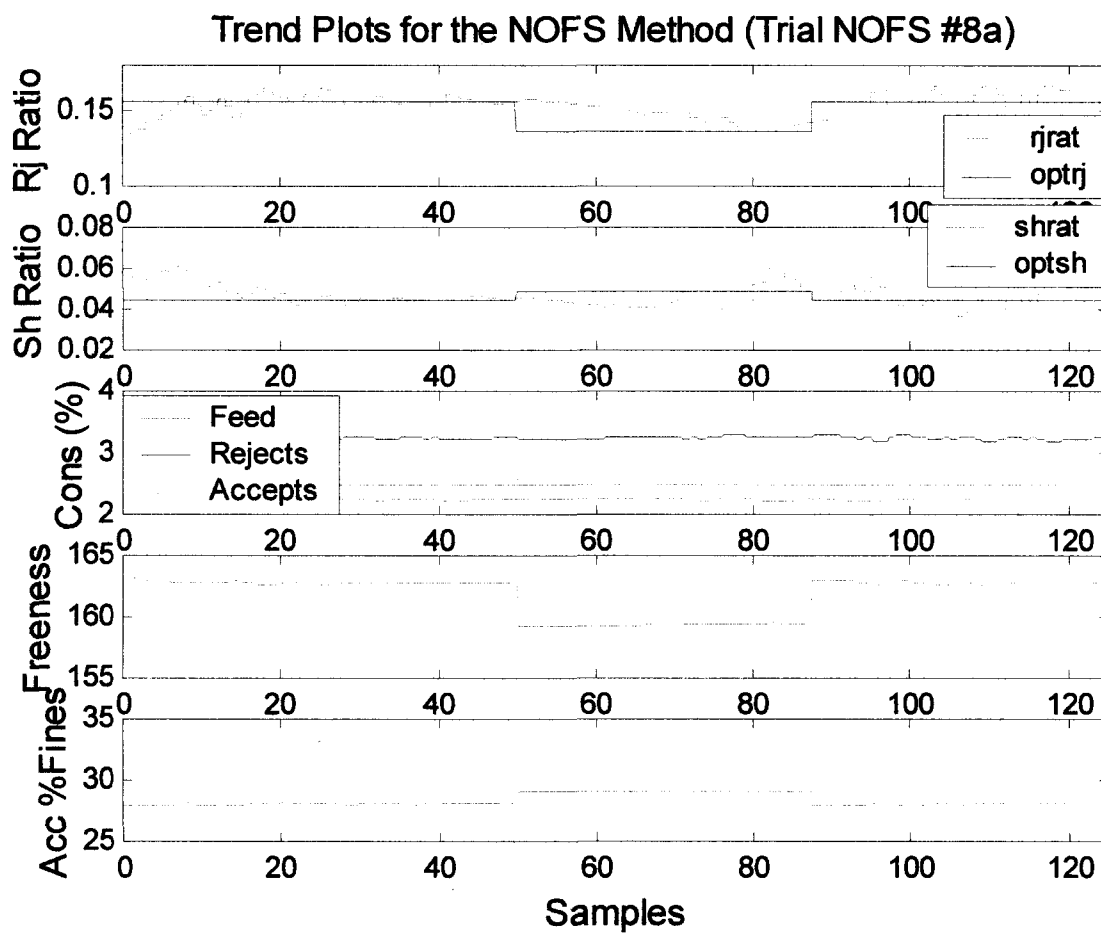


Figure 3.29(b): Trend plots for the operating variables for the simulation Trial NOFS #8a.

Changes to the feed consistency was the final type of disturbance used to test the either simplex method. Changes to the feed consistency can cause significant disruptions to the operation of the refined rejects pressure screen. Without any control technique in place, an increase in the feed consistency would lead to increases in the consistency of the accepts and rejects streams. Increases to the consistency of the rejects stream can lead to plugging of the screen and prevent the screen from efficiently separating the fibres. A decrease to the feed consistency can lead to the presence of excess water in the accepts stream, and a greater energy requirement to remove the water downstream from the pressure screen.

The simulations testing the ability of the simplex to handle the feed consistency disturbance are illustrated in Figures 3.30 through 3.34. In all of the simulations where a feed consistency disturbance was applied, the feed consistency was increased at the 50<sup>th</sup> sample from 2.48% to 2.78%. The feed consistency was then returned to 2.48% at the 88<sup>th</sup> sample. Figures 3.30 and 3.31 illustrate the simulations run using the EOFS method. Some of the operating variables for Figures 3.30 and 3.31 are as follows:

- starting simplex size = 0.005
- minimum simplex size = 0.000625
- maximum simplex size = 0.011

Figures 3.30(a) and (b) illustrate Trial EOFS #6, which is a simulation that uses an expansion factor of 1.25 and a contraction factor of 0.8 against a feed consistency disturbance. The results of this simulation are very good. Not only does the simplex track the moving optimum, but also corrects the consistency when it moves out of its acceptable range. When the disturbance entered the simulation the reject ratio increased to prevent the screen from plugging. Shortly after that, the shower ratio increased to help bring the rejects consistency back to a more acceptable value. The rejects consistency did remain above its original value, but was within the acceptable range set in the cost equation. The acceptable range for the rejects consistency in the cost equation was between 2.7% and 3.5%.

Figures 3.31(a) and (b) illustrate the simulation Trial EOFS #7. EOFS #7 is a simulation with noise added and a feed consistency disturbance. Similar to the previous EOFS simulations that included noise, the simplex was unable to search out either optimum ratio or make any correction to the rejects consistency.

Therefore, if the sequential simplex method is used as the control and optimization technique for the refined rejects pressure screen smaller expansion and contraction factors should be used.

Figures 3.32 through 3.35 illustrate the simulations using the NOFS method. Some of the operating variables used for the simulations shown in Figures 3.32 through 3.35 are as follows:

- starting simplex size = 0.005
- minimum simplex size = 0.001
- maximum simplex size = 0.014

Figures 3.32(a) and (b) display the trend plots for the simulation Trial NOFS #9. This simulation uses the NOFS method with no ratio restrictions placed on the expansion or contraction criteria. The maximum reject consistency is 3.5% and the minimum is 2.7%. Upper and lower limits have been applied to the reject consistency to prevent plugging and excessive watering, using the cost equation. In Figure 3.32(b), the rejects consistency is oscillating around 3.1% during the first 50 samples. During the next 38 samples, the disturbance causes the rejects consistency to increase to almost 3.4%. During this period the simplex responds with an increase in the reject and shower ratio's, resulting in a small decrease in the rejects consistency. Increasing the penalty in the cost equation for a high rejects consistency would provide a stronger response from the simplex for a high rejects consistency, but could adversely affect the response to other disturbances. From the 88<sup>th</sup> sample to the end of the simulation, the feed consistency returned to its original value causing the rejects consistency to return to pre-disturbance levels.

Figures 3.33(a) and (b) display the trend plots for the simulation Trial NOFS #10. This simulation uses the NOFS method and the constraint on the shape of the simplex. The results of this simulation are similar to the previous one, except the oscillations are smaller. Trial NOFS #10 shows good results when responding to and correcting for disturbances in the screen. After the disturbance is entered into the simulation at the 50<sup>th</sup> sample the reject and accept consistencies rise. The shower ratio responds with an increase, causing the shower flow to increase. The results of the increased shower flow is seen by a decrease in the reject and accept consistency. Within approximately 25 samples of the disturbance entering the simulation both the reject and accept consistencies are returned to their values prior to the disturbance. At the 88<sup>th</sup> sample the feed consistency drops back to 2.48%. The NOFS method responds to this by decreasing both the reject and shower ratio's. This response helps to return both the reject and accept

consistencies back to their original value. The response in this simulation is a little slower than in the previous simulation. It seems that the constraints on the shape of the simplex may slow the ability of the simplex to respond to changes. The constraints were put in to prevent the simplex from becoming too elongated in one direction, but in the simulations without the constraint, the results have been better and the simplex did not become elongated.

Figures 3.34(a) and (b) display the trend plots for the simulation Trial NOFS #11. This simulation uses the NOFS method with the shape constraint in place, and a target fibre distribution instead of the fibre separation term in the cost equation. Although the simplex searches out the optimum in a quick manner its progression is much different than the previous simulations. The reject consistency is corrected quickly after each disturbance. This action doesn't seem to create any problems, but does leave an impression of an unstable control method. It is expected that at some time after the disturbance has been corrected that the simplex will settle down and shrink towards an optimum.

Figures 3.35(a) and (b) illustrates Trial NOFS #12, which includes the shape constraint and noise in the simulations. There is not much response to the disturbance in this simulation. Similar to other simulations with noise added, it may be caused by the noise on the measurements of the reject and accept consistencies. If the value of the measurement isn't accurate, then the movement of the simplex will be slow and inaccurate. The noise may cause each simplex movement to be in opposite directions, therefore preventing the simplex from making any progress in the right direction.

The standard deviation of the noise was arbitrarily chosen and may be greater than the actual noise encountered from the sensors in the plant. If the noise in the plant is less than the simulated noise, the action of the simplex would resemble the results of Figures 3.35(a) and (b). If the actual amount of noise in the plant is greater than the simulated noise, the detection of a disturbance by the simplex would be difficult, and the penalty for a poor rejects consistency would need to be increased in the cost equation. Increasing the penalty for the rejects consistency in the cost equation can lead to large cost value for small errors in the rejects consistency.



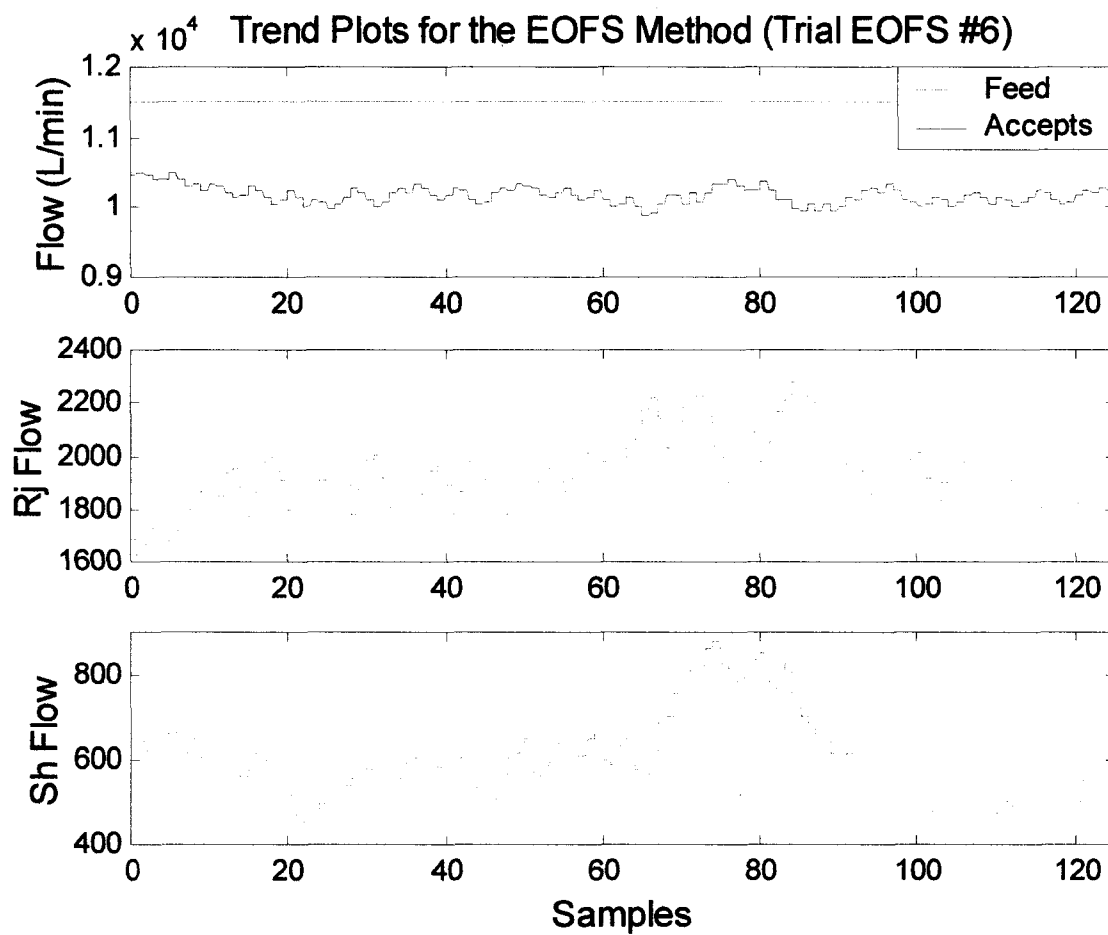


Figure 3.30(a): Trend plots for the flows of each stream for the simulation Trial EOFS #6.

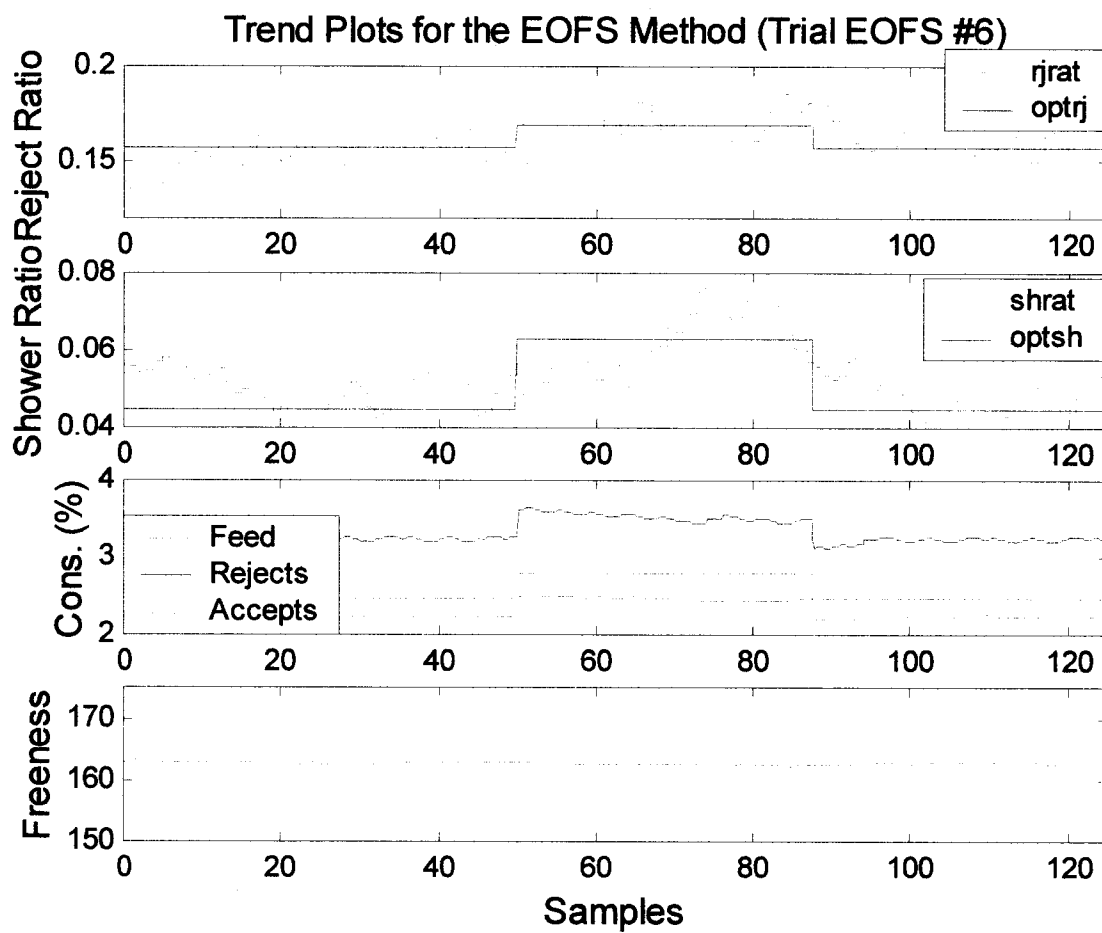


Figure 3.30(b): Trend plots for the flows of each stream for the simulation Trial EOFS #6.

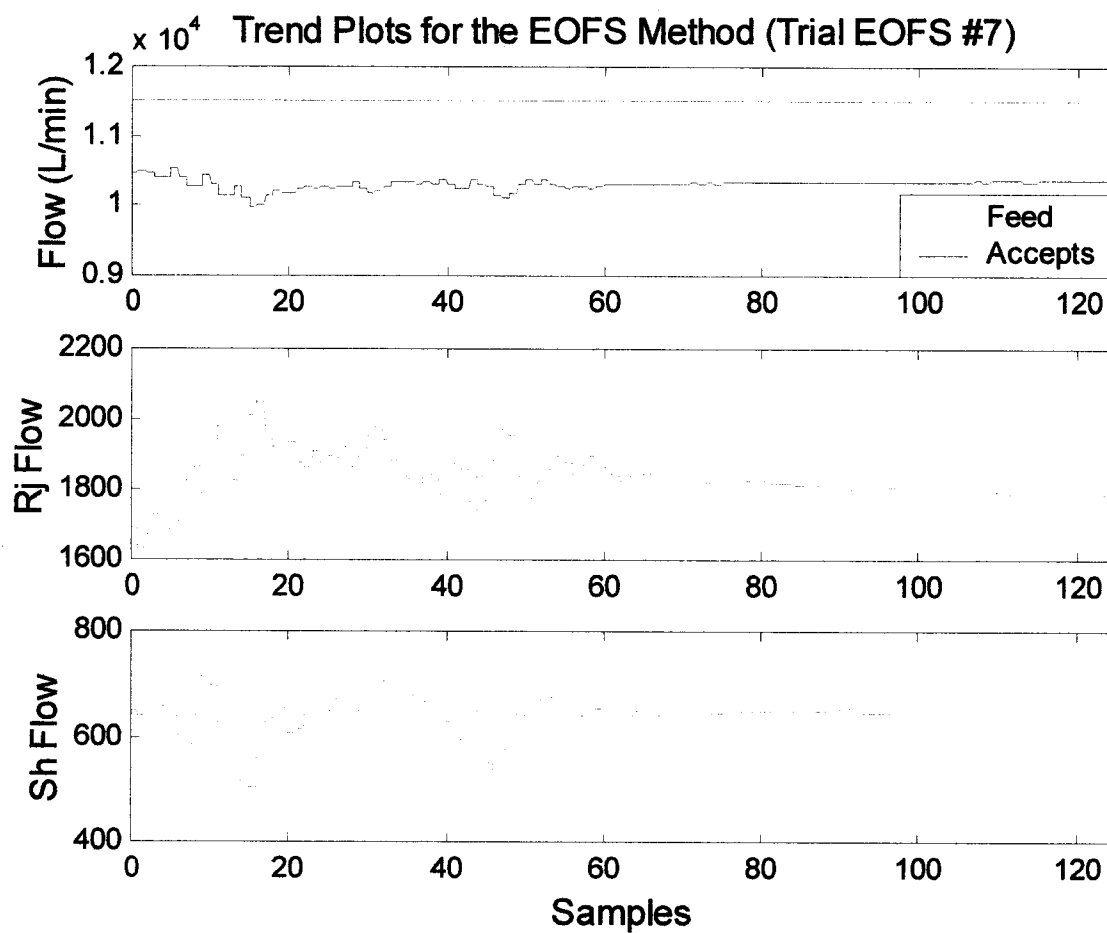


Figure 3.31(a): Trend plots for the flows of each stream for the simulation Trial EOFS #7.

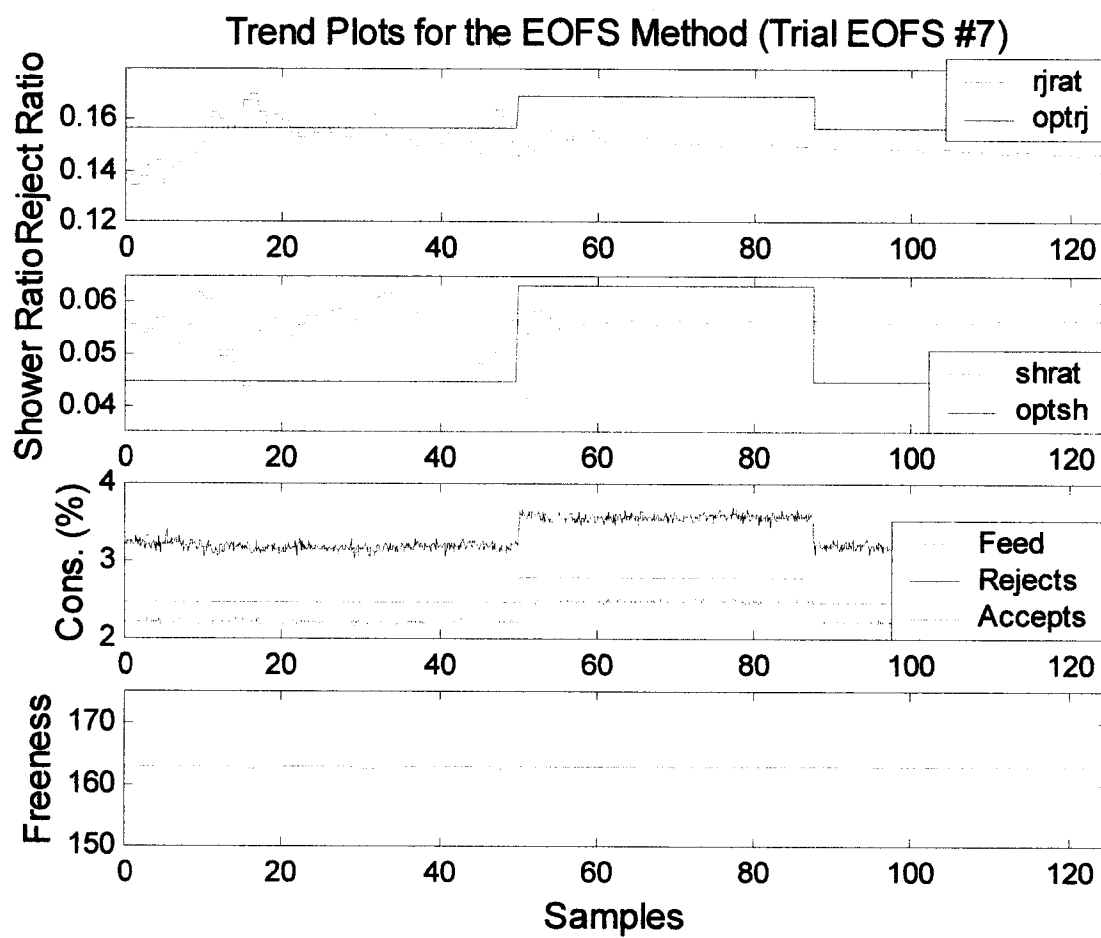


Figure 3.31(b): Trend plots for the operating variables for the simulation Trial EOFS #7.

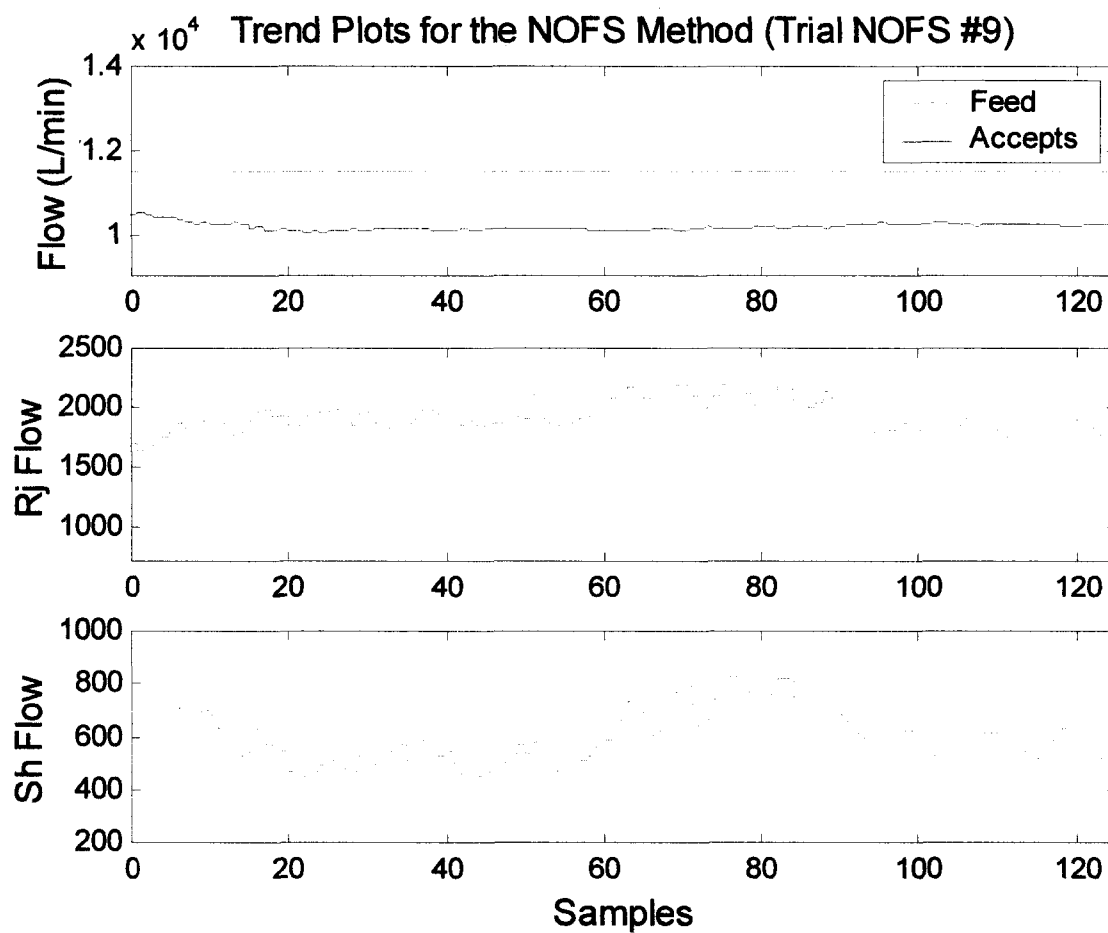


Figure 3.32(a): Trend plots for the flows of each stream for the simulation Trial NOFS #9.

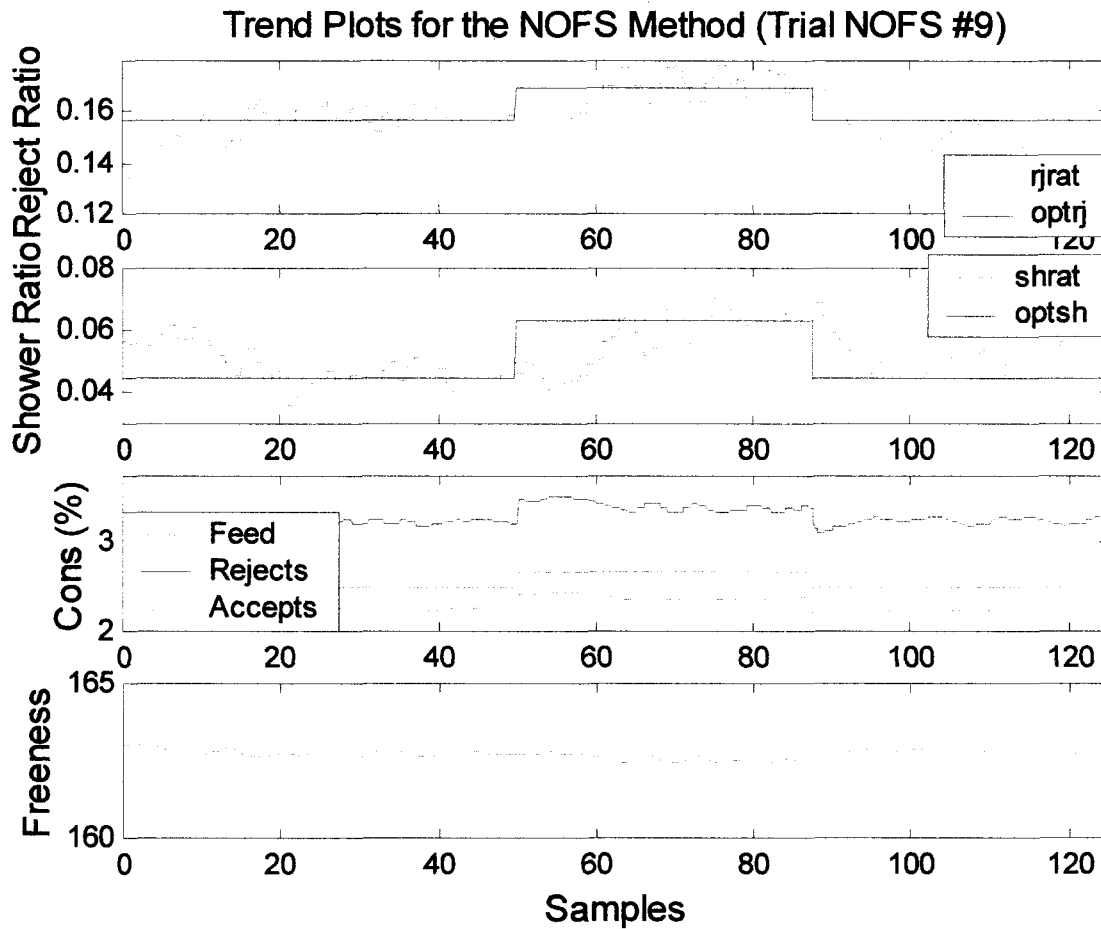


Figure 3.32(b): Trend plots for the operating variables for the simulation Trial NOFS #9.

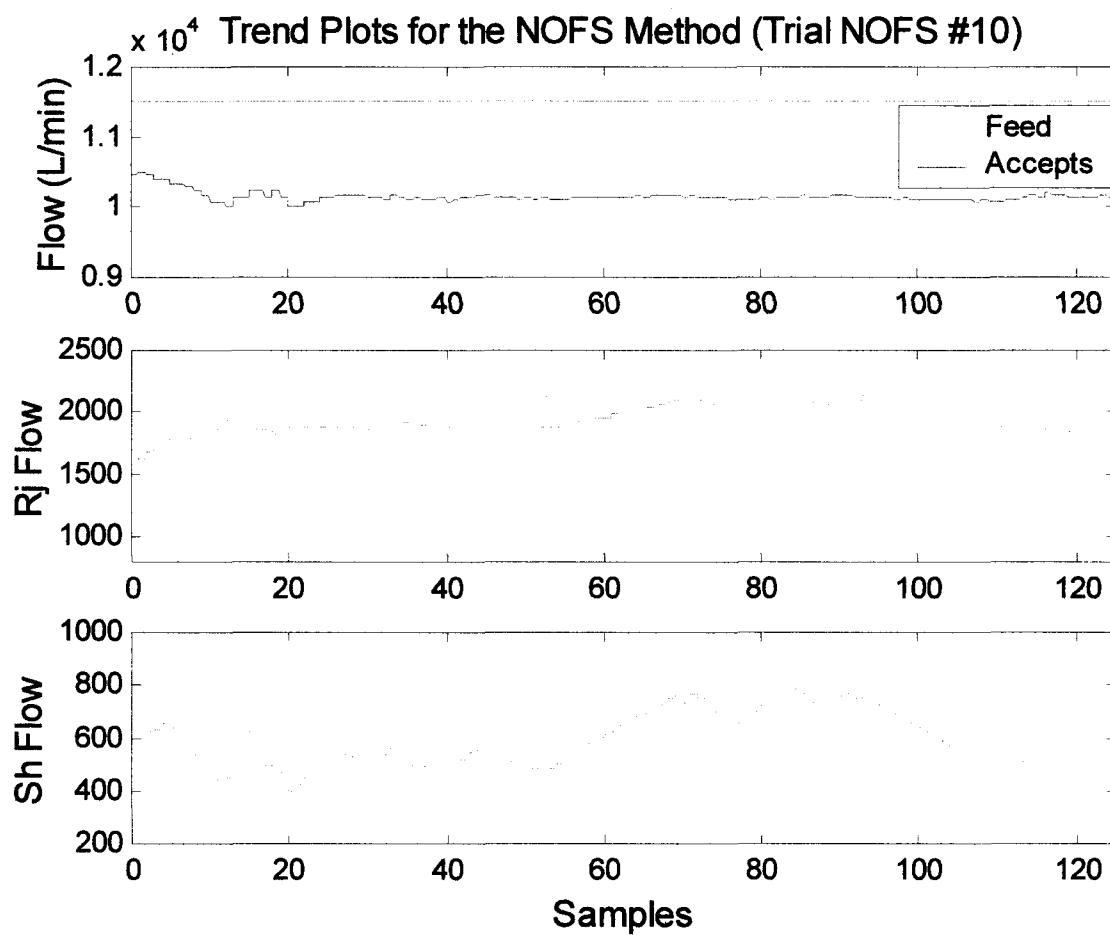


Figure 3.33(a): Trend plots for the flows of each stream for the simulation Trial NOFS #10.

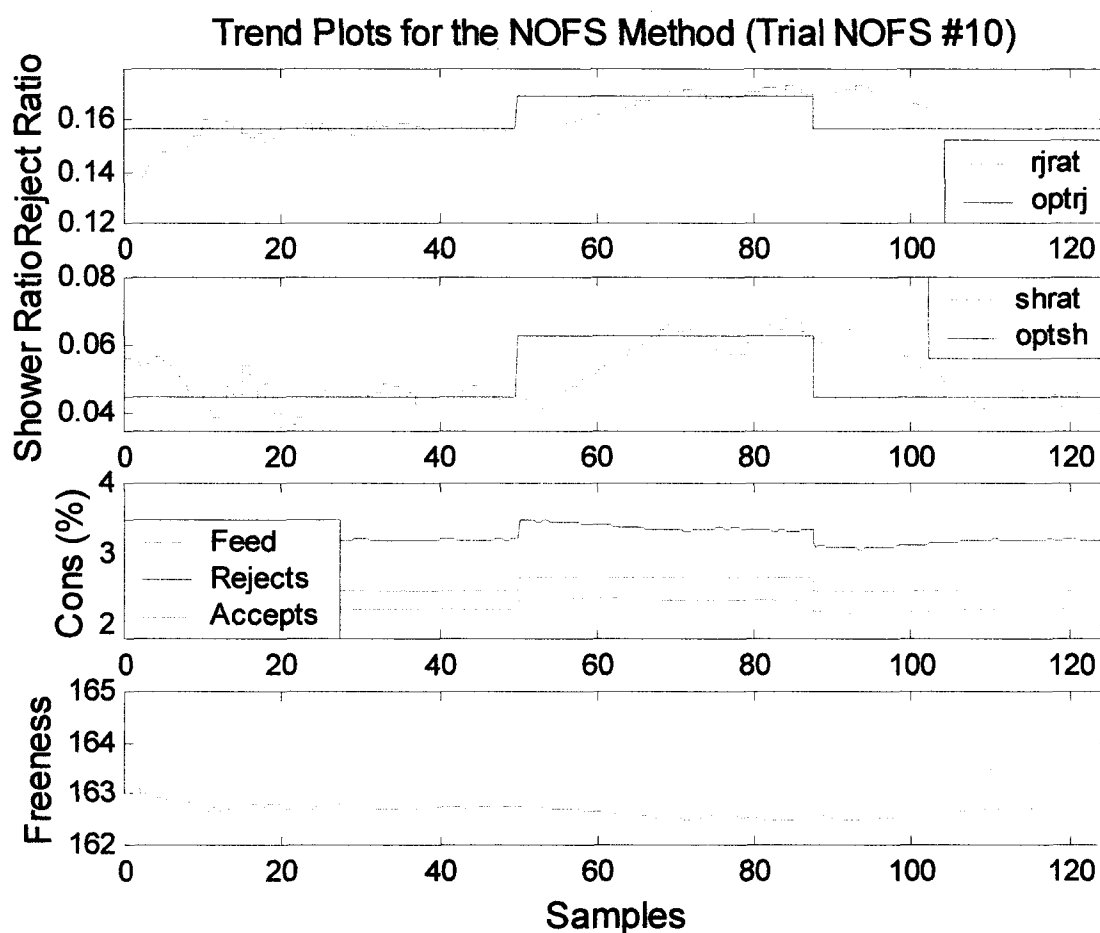


Figure 3.33(b): Trend plots for the operating variables for the simulation Trial NOFS #10.



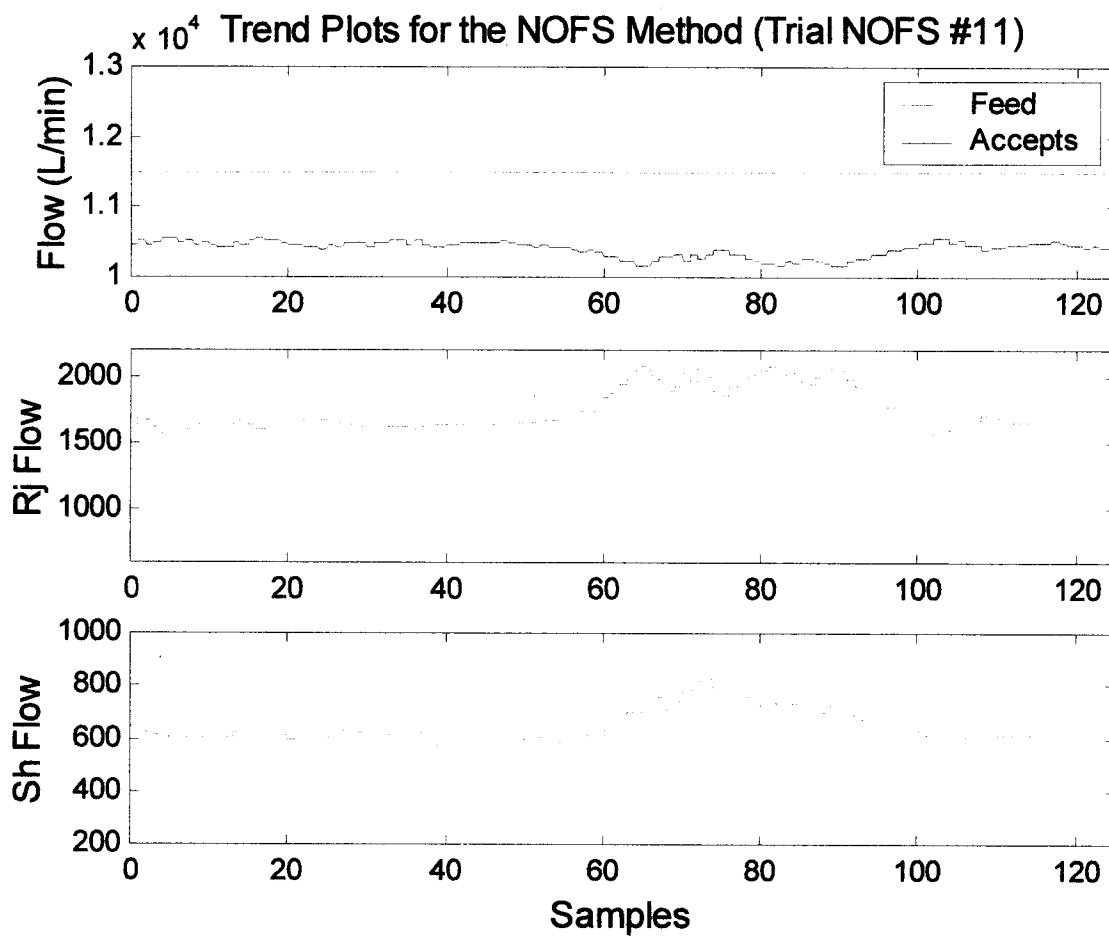


Figure 3.34(a): Trend plots for the flows of each stream for the simulation Trial NOFS #11.

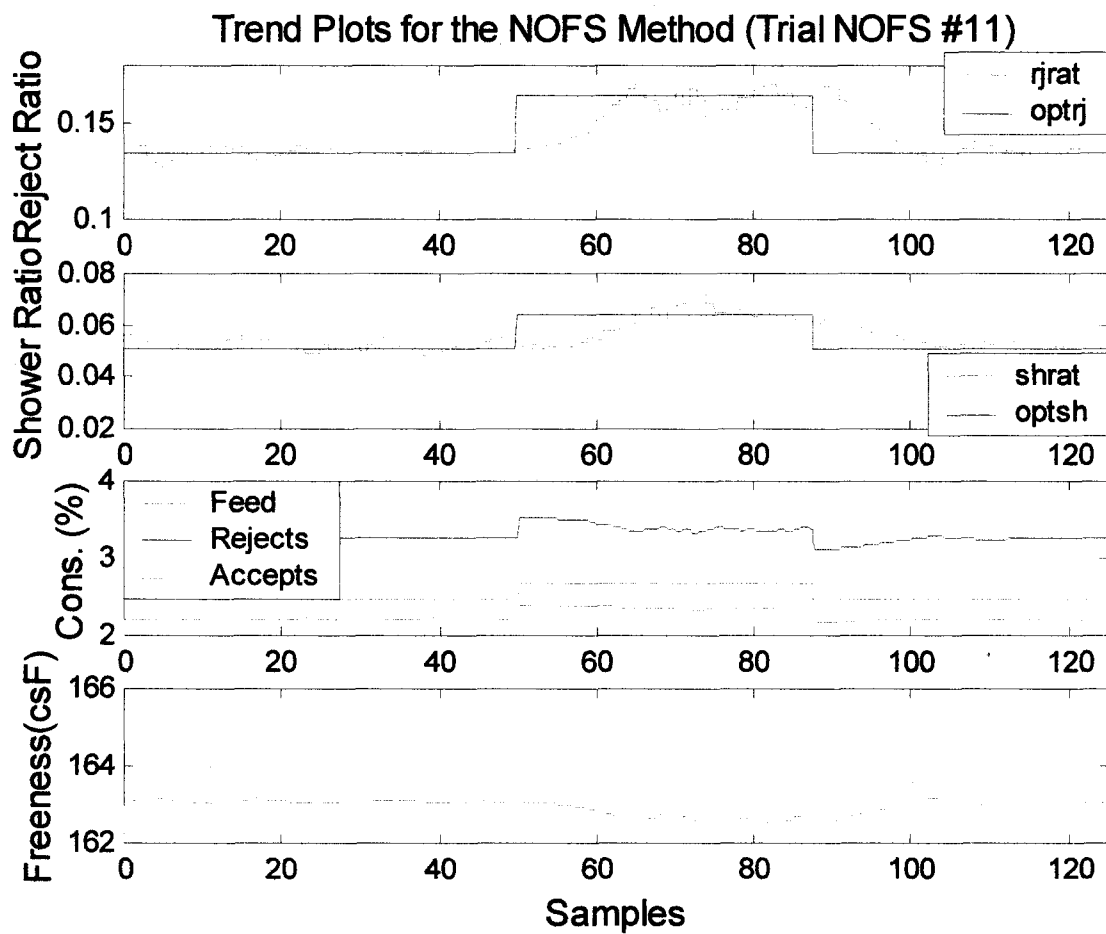


Figure 3.34(b): Trend plots for the operating variables for the simulation Trial NOFS #11.

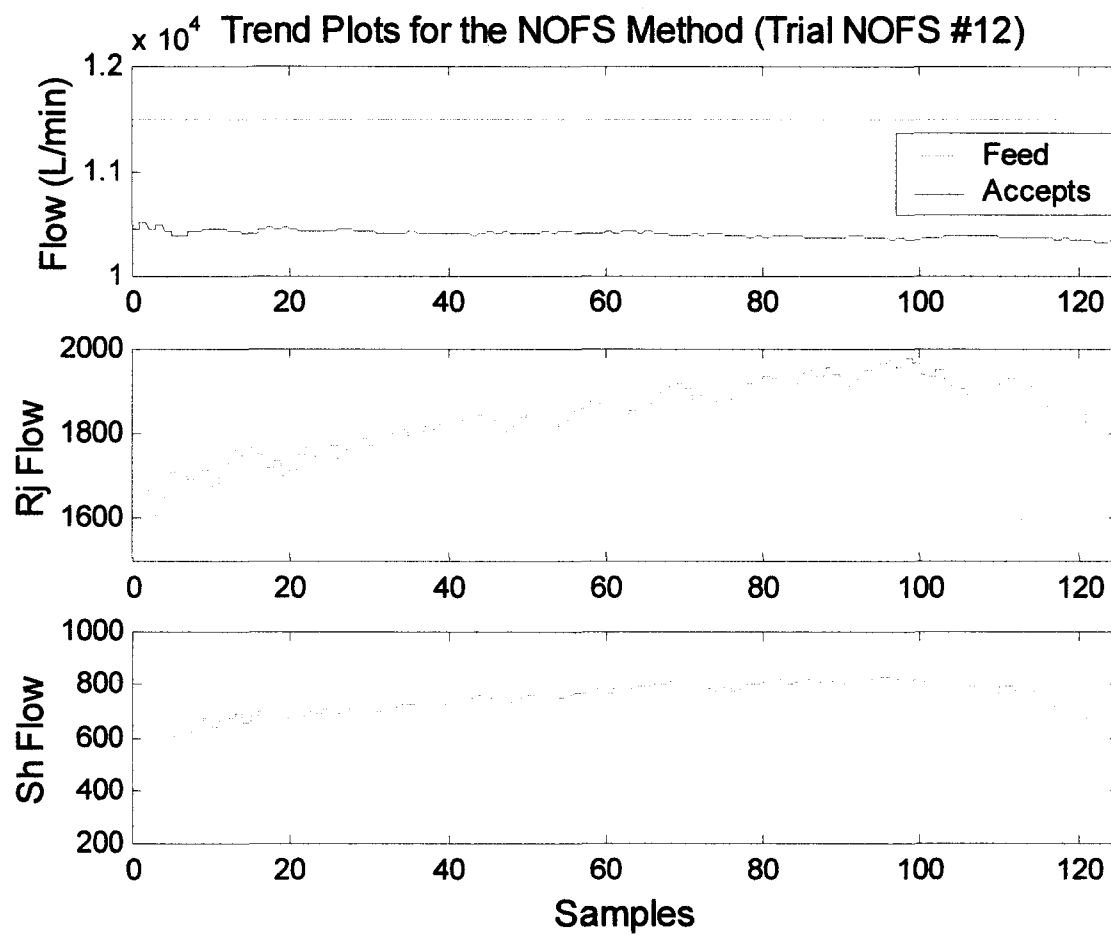


Figure 3.35(a): Trend plots for the flows of each stream for the simulation Trial NOFS #12.

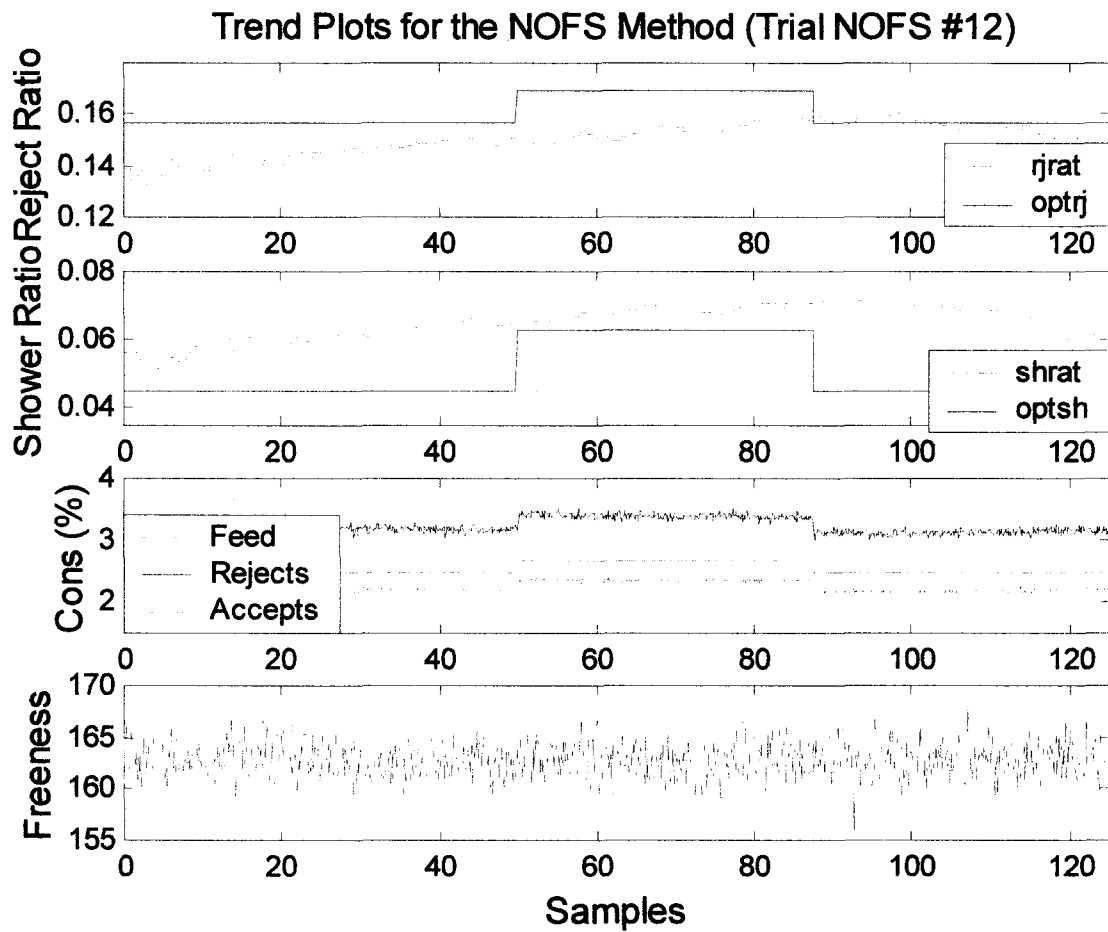


Figure 3.35(b): Trend plots for the operating variables for the simulation Trial NOFS #12.

### **3.5 Comparing the Optimum Following Simplex Methods during Disturbances**

The response to a feed flow disturbance is handled well by both simplex methods at all of the different settings. It does not seem to matter how large or small the simplex is, or how it expands and contracts, each method worked well. The most important factor with the feed flow disturbance was to determine if the either the reject or shower ratio changed as a result of the feed flow disturbance. As expected, in all cases the reject and shower ratios were unaffected by the disturbance, allowing for changes to the reject and shower flows only. Although all simulations were successful, some were better than others. Using the EOFS method the simplex that expanded by 1.25 and contracted by 0.8 settled better than the simplex that expanded by 2.0 and contracted by 0.5. This result was expected because rapid changes in the simplex using the factors 2.0 and 0.5 would be problematic when the simplex was already somewhat large or very small. Using the small expansion and contraction factors is preferred.

In a comparison of the NOFS Trials #1 and #2, both were very good at searching the optimum as well as settling down quickly. Trial #2 was slightly better, but Trial #1 worked surprisingly well without the shape constraint applied.

Since a feed flow disturbance has little effect on the consistency or fibre distribution of any of the streams entering or exiting the screen, the noise applied in the feed flow disturbance simulations should have little affect. Although noise does affect measurements of the consistencies and freeness, it should not affect the ability of the simplex to search for the optimum ratios. Unfortunately, both the EOFS and NOFS methods struggle when noise is present.

The ability of the EOFS and NOFS methods to search for the optimum reject and shower ratios during a period when a feed fibre distribution disturbance occurs is illustrated in Figures 3.23 through 3.28. Both of the simplex methods worked very well under the presence of a disturbance to the feed fibre distribution when noise wasn't added to the simulation. Trial NOFS #5 shows the best results. The NOFS method along with the conditions in Trial NOFS #5 seem to produce the best results. All of the simulations without noise were able to search and track the optimum reject and shower ratios. The simulation with the shape constraint works, but it's slower than Trial NOFS #5. The simulations with the noise added produced

very poor results. As mentioned earlier, the noise is the problem in those simulations, and that a running average would probably fix it, but a shorter sampling period is required.

Under the presence of a disturbance to the feed consistency, Trial NOFS #9 illustrates the simplex method most capable of responding to the disturbance. Although the correction to the consistency was not complete, it responded quickly by adjusting both ratios. The simplex responds to the disturbances in both directions and partially corrects for the deviations in the consistencies. Other than the simulations with noise added, all simulations were able to correct for the changes to the consistency. Unfortunately, each of the simulations with noise added were unable to perform as well as those without noise.

## Conclusions

In the preceding chapters it was shown that the sequential simplex method and the Nelder and Mead method can be converted and modified from a limited search method to one that can search a system that has a moving optimum.

The results show two very distinct response patterns common to both the NOFS and the EOFS methods. The NOFS method offers more flexibility when searching for the optimum than the EOFS. The most effective search method appears to be the NOFS method when no constraints are placed on the expansion and contraction of the simplex, illustrated in Trials NOFS #1, #5 and #9. The danger with this method is the simplex can grow in one direction, becoming very elongated in one direction, and failing to quickly respond to shifts in the optimum that are perpendicular to the elongated direction of the simplex. A more conservative approach is shown in Trials NOFS #2, #6 and #10, where constraints are placed on the expansion and contraction of the simplex.

Without the presence of noise in the system both methods were able to respond to disturbances effectively. Unfortunately, in the presence of noise both methods failed to search both the optimum reject and shower ratios. To improve the ability when running either simplex method in the presence of noise, some averaging technique could be used to dampen the effect of a noisy measurement. For the refined rejects pressure screen some of the measurements are provided by the PQM. The PQM measures using four ports at intervals of ten minutes. Therefore, each measurement dedicated to the refined rejects pressure screen is only provided every 40 minutes. This measurement frequency is not sufficient if applying an averaging technique to the measurements of the refined rejects pressure screen. Dedicating all four ports of the PQM to the refined rejects pressure screen would improve the average value of each of the measurements. Although this would improve the results, the average value of five samples would still be based on measurements ranging 40 minutes. This method would be sufficient if the goal was to maintain a steady state of a process that experienced very few disturbances, but would still fail to respond to any disturbances.

The maximum fibre separation performance equation requires knowledge of the fibre distribution of the accepts and rejects streams, while the target fibre distribution only requires knowledge of the rejects

stream. With the high cost PQM units and the lengthy sampling period, it is clearly an advantage to optimize the refined rejects pressure screen using the a performance equation based on a target fibre distribution.

The amount of noise added in the simulation needs to be confirmed with the actual amount in the plant. If the actual noise is less than shown here, both of the simplex methods will respond more effectively to disturbances.

Therefore, even if this method doesn't work on the refined rejects pressure screen, it has shown that it may be very useful on other processes in many industries.



## **Recommendations**

1. Decrease the length of the sampling period by dedicating a PQM to the Refined Rejects Pressure Screen.
2. Further investigate the minimum and maximum simplex size limits.
3. Further investigate the measurement noise of the refined rejects pressure screen to better quantify the standard deviation of the noise.
4. Examine other processes and industries where the sampling frequency is improved over the refined rejects pressure screen.
5. Determine a more effective equation for the prediction of Freeness in the simulation code.
6. Implement the methods developed in a industrial setting.

## References

- [1] Leask, R.A., Pulp and Paper Manufacture, vol. 2: Mechanical Pulping, 3<sup>rd</sup> ed. (Canada: Joint Textbook Committee of the Paper Industry, 1987) 181.
- [2] Bowater Process Flowsheet and Process and Instrumentation Diagrams.
- [3] Wakelin, Ross F., "Prediction of fractionation efficiency for pressure screens," Appita Journal 50.4 Jul. (1997): 295-300.
- [4] Gooding, R.W., R.J. Kerekes, "Derivation of Performance Equations for Solid-Solid Screens," The Canadian Journal of Chemical Engineering Oct. 1989: 801.
- [5] Hooper Screen Manual Index.
- [6] Jamie Puumala, "Screen Dimensions," 12 March 2001, personal email (13 March 2001).
- [7] Nelson, George L., "The screening quotient: a better index for screening performance," Tappi 64.5 May. (1981): 133-134.
- [8] Gooding, R.W., R.J. Kerekes, "Consistency changes caused by pulp screening," Tappi Journal Nov. 1992: 109.
- [9] Allison, B.J., J.A. Olson, "Optimization of Multiple Screening Stages for Fibre Length Fractionation: Two-Stage Case," Journal of Pulp and Paper Science 26.3 Mar. (2000): 114.
- [10] Olson, J., B. Allison, N. Roberts, "Fibre Length Fractionation Caused by Pulp Screening. Smooth-Hole Screen Plates," Journal of Pulp and Paper Science 26.1 Jan. (2000): 12-16.
- [11] Olson, J., N. Roberts, B. Allison, R. Gooding, "Fibre Length Fractionation Caused by Pulp Screening," Journal of Pulp and Paper Science 24.12 Dec. (1998): 395.
- [12] Olson, J.A., "Fibre Length Fractionation Caused by Pulp Screening, Slotted Screen Plates," Journal of Pulp and Paper Science 27.8 Aug. (2001): 255-261.
- [13] Liu, X.A., P. Whiting, H. Pande, D.N. Roy, "The Contribution of Different Fractions of Fines to Pulp Drainage in Mechanical Pulps," Journal of Pulp and Paper Science 27.4 Apr (2001): 139.
- [14] Edgar, T.F. and D.M. Himmelblau, Optimization of Chemical Processes (United States: McGraw-Hill, Inc., 1988) 4.
- [15] Beveridge G.S.G., and R.S. Schechter, Optimization: Theory and Practice (United States: McGraw-Hill, Inc., 1970) 367.
- [16] Nelder, J.A. and R. Mead, "A simplex method for function minimization," Computer Journal 7:4 (1965): 308.
- [17] Press, William H., and Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. Numerical Recipes in C: The Art of Scientific Computing, 2<sup>nd</sup> ed. (United States: Cambridge University Press, 1992): .
- [18] Technical Specifications, PQM 400 Manual Date:891207.
- [19] Xiong, Q. and A. Jutan, "Continuous Optimization using a Dynamic Simplex Method," (2000).
- [20] Natarajan, K., A.F. Gilbert, A. Yousef, "PID controller tuning based on finite frequency response data," Pulp & Paper Canada 99:4 (1998): 60.

## Appendix A: simplex1.c

```
#define GET_PSUM for(j=1;j<=ndim;j++) {for(sum=0.0,i=1;i<=mpts;i++)sum+=p[i][j];psum[j]=sum;}

/***** performance function() *****/
* The performance function calculates the cost function from the target
* fibre fractions and the fibre fractions from the simulation. The
* equation is used to find the minimum.
*****/
float func(struct modeldat *md)
{
    float *y, *x, *k, cost;

    y = md->y;
    x = xglobal;
    k = kglobal;

    /*calculate the value of the performance equation.*/
    cost = 0.0;
    cost = 500*(RJ RAT-X1SS)*(RJ RAT-X1SS) + 500*(SHRAT-X2SS)*(SHRAT-X2SS);

    return(cost);
}

/***** simplex() *****/
* The simplex function is a controller designed to find the optimum operating
* conditions of the dynamic model provided. The three points of the starting
* simplex are calculated and evaluated using the performance equation. A new
* simplex is determined based on the reflection of the old simplex away from
* the worst point.
* If the best point remains unchanged for 4 consecutive reflections, the size
* of the simplex is reduced.
*****/
void simplex(struct timing *t,struct modeldat *md)
/*Need the current reject and shower flow rates, and their corresponding fibre fractions.*/
{
    float value;
    float *y,*x,*ss,*k;
    struct simp *simp;
    int i,j,mpts,ndim,simppt;
    float sum, *psum, *ct, *xnew, *ynew;
    float facrj, facsh, size, rjdif, shdif;
    static int rejnumber, expand, ilo, inhi, ihi;
    static float best[4],a,b,fhold, phold1, phold2;
    float rejected;
    float ctopilo, ctopinhi, ctopihi, minlength, maxlength;
    float fin,med,cor;
    fin=1;
    med=1;
    cor=1;

    x = xglobal;
    ss = ssglobal;
    k = kglobal;
}
```

```

/*call the current values of the flow rates and fractions.*/
y = md->y;      /*the address of the states in the modeldat structure.*/
/*x = md->x;      /* " " " " inputs in the modeldat structure.*/

ndim = md->k[2]; /*the number of coordinates in each pt of the simplex.*/
mpts = ndim+1;
ilo = 1;
inhi = 1;
ihi = 1;

psum=vector(1,ndim);
ct=vector(1,ndim);
xnew=vector(1,ndim);
ynew=vector(1,ndim);

/*calculate the changes to be made to the simplex.*/
a = SIMPSIZE/SIMPNP/sqrt(2)*(sqrt(SIMPNP+1)+SIMPNP-1);
b = SIMPSIZE/SIMPNP/sqrt(2)*(sqrt(SIMPNP+1)-1);

if (p[1][1] <= 0 ) simppt=1;
else if (p[2][1] <= 0) simppt=2;
else if (p[3][1] <= 0) simppt=3;
else simppt=4;

if (simppt == 1){
    /*evaluate the performance equation for the first simplex point,
    and place the coordinates and perf in the simplex structure.*/
    value = func(md);

    /*indirect the pt number, inputs, performance value, best point.*/
    p[1][1] = RJRAT;
    p[1][2] = SHRAT;
    fv[1] = value;

    /* Determine the factors for the direction of the 2nd point of the
    starting simplex.*/
    if (p[1][1] > (LLRJRAT+ULRJRAT)/2) facrj = -1.0;
    else facrj = 1.0;

    if (p[1][2] > (LLSHRAT+ULSHRAT)/2) facsh = -1.0;
    else facsh = 1.0;

    /*calculate the coordinates of the second point of the simplex.*/
    md->x[0] = RJRAT + facrj*a;
    md->x[1] = SHRAT + facsh*b;
    /*printf("RJRAT %f SHRAT %f ", md->x[0], md->x[1]);*/
} else if (simppt == 2) {
    /*evaluate the performance equation for the second point.*/
    p[2][1] = RJRAT;
    p[2][2] = SHRAT;
    value = func(md);

    /*indirect the pt number, inputs, performance value, best point.*/
    fv[2] = value;

```

```

/* Determine the factors for the direction of the 3rd point of the
   starting simplex.*/
if (p[1][1] > (LLRJRAT+ULRJRAT)/2) facrj = -1.0;
else facrj = 1.0;

if (p[1][2] > (LLSHRAT+ULSHRAT)/2) facsh = -1.0;
else facsh = 1.0;

/*calculate the coordinates of the third point of the simplex.*/
md->x[0] = p[1][1] + facrj*2*b;
md->x[1] = p[1][2] + facsh*a;

} else if (simppt == 3) {
/*evaluate the performance equation for the current point.*/
p[3][1] = RJRAT;
p[3][2] = SHRAT;
value = func(md);

/*indirect the pt number, inputs, performance value, best point.*/
fv[3] = value;

GET_PSUM

/* Determine the ranking of each point in the simplex. The lowest (or
   best) will be labelled "ilo", the middle pt "inhi", and the high
   (or worst) pt "ihi". */
ilo=1;
ihi = fv[1]>fv[2] ? (inhi=2,1) : (inhi=1,2);
for (i=1;i<=mpts;i++){
    if (fv[i] <= fv[ilo]) ilo=i;
    if (fv[i] > fv[ihi]) {
        inhi = ihi;
        ihi = i;
    } else if (fv[i] > fv[inhi] && i != ihi) inhi=i;
}
/*calculate the coordinates of the new point in the simplex.*/
/*Reflect the simplex away from the worst point.*/
for (j=1;j<=ndim;j++) {
    xnew[j]=psum[j]-2.0*p[ihi][j];
}
p[ihi][1]=xnew[1];
p[ihi][2]=xnew[2];
rejnumber = ihi;

/*Update the Valve settings based on simplex movement.*/
md->x[0] = xnew[1];
md->x[1] = xnew[2];

expand=0;

} else {
/*evaluate the performance equation for the current point.*/
/*Find the cost of the current valve settings.*/
value = func(md);
rejected = fv[rejnumber]; /*Holder for the rejected cost.*/

```

```

fv[rejnumber] = value; /*place the new cost into the rejected row.*/

/*if the expanded pt is worse than the previous pt.*/
if(expand==1 && value>=rejected) {

    /*reverse the expansion*/
    p[rejnumber][1]=phold1;
    p[rejnumber][2]=phold2;
    fv[rejnumber]=fhold;
    /*printf("REVERSE \n");*/
    /*printf("RJ Pts %f %f %f \n",p[1][1],p[2][1],p[3][1]);*/
    /* Determine the ranking of each point in the simplex. The
    lowest (or best) will be labelled "ilo", the middle pt "inhi",
    and the high (or worst) pt "ihi". */
    ilo=1;
    ihi = fv[1]>fv[2] ? (inhi=2,1) : (inhi=1,2);
    for (i=1;i<=mpts;i++){
        if (fv[i] <= fv[ilo]) ilo=i;
        if (fv[i] > fv[ihi]) {
            inhi = ihi;
            ihi = i;
        } else if (fv[i] > fv[inhi] && i != ihi) inhi=i;
    }
    GET_PSUM
    /*and reject the worst pt from the unexpanded simplex.*/
    /*Reflect the simplex away from the worst point.*/
    for (j=1;j<=ndim;j++) {
        xnew[j]=psum[j]-2.0*p[ihi][j];
    }
    p[ihi][1]=xnew[1];
    p[ihi][2]=xnew[2];
    rejnumber = ihi;
    expand=0;
} else {

    GET_PSUM

    /* Determine the ranking of each point in the simplex. The lowest (or
    best) will be labelled "ilo", the middle pt "inhi", and the high
    (or worst) pt "ihi". */
    ilo=1;
    ihi = fv[1]>fv[2] ? (inhi=2,1) : (inhi=1,2);
    for (i=1;i<=mpts;i++){
        if (fv[i] <= fv[ilo]) ilo=i;
        if (fv[i] > fv[ihi]) {
            inhi = ihi;
            ihi = i;
        } else if (fv[i] > fv[inhi] && i != ihi) inhi=i;
    }

    /*Place the best pt into an extra matrix used to prevent the simplex
    from pivoting the best vertex for more than 4 iterations.*/
    best[4] = best[3];
    best[3] = best[2];
    best[2] = best[1];
    best[1] = fv[ilo];

```

```

/*calculate the coordinates of the new point in the simplex.*/
if (rejnumber==ihi) {
    /*newest pt is worse than most recent rejected, reject the 2nd worst*/
    for (i=1;i<=ndim;i++) xnew[i]=psum[i]-2.0*p[ihi][i];
    p[ihi][1]=xnew[1];
    p[ihi][2]=xnew[2];
    rejnumber = ihi;

} else if (best[1]==best[2]&&best[2]==best[3]&&best[3]==best[4]&&best[1]!=0.0){
    ct[1] = psum[1]/3;
    ct[2] = psum[2]/3;

    ctopilo=sqrt((ct[1]-p[iilo][1])*(ct[1]-p[iilo][1])+(ct[2]-p[iilo][2])*(ct[2]-p[iilo][2]));
    ctopinhi=sqrt((ct[1]-p[iinhi][1])*(ct[1]-p[iinhi][1])+(ct[2]-p[iinhi][2])*(ct[2]-p[iinhi][2]));
    ctopihi=sqrt((ct[1]-p[iihi][1])*(ct[1]-p[iihi][1])+(ct[2]-p[iihi][2])*(ct[2]-p[iihi][2]));

    if (ctopilo<ctopinhi && ctopilo<ctopihi) minlength=ctopilo;
    else if (ctopinhi<ctopilo && ctopinhi<ctopihi) minlength=ctopinhi;
    else minlength=ctopihi;

    if (minlength>SIMPMIN && ctopihi/ctopilo<1.7 && ctopinhi/ctopilo<1.7) {
        /*simplex is cycling around one pt, shrink the simplex.*/
        for (i=1;i<=ndim;i++){
            xnew[i]=0.25*(psum[i]+p[iilo][i]);
        }
        p[iilo][1]=xnew[1];
        p[iilo][2]=xnew[2];

        best[1]=0.0;    /*reset the best matrix*/
        rejnumber = iilo;
    } else {
        /*Reflect the simplex away from the best point.*/
        for (j=1;j<=ndim;j++) {
            xnew[j]=psum[j]-2.0*p[iilo][j];
        }
        p[iilo][1]=xnew[1];
        p[iilo][2]=xnew[2];
        rejnumber = iilo;
        expand=0;
        best[1] = 0.0;
        /*printf("No shrink \n");*/
    }
}

} else if (rejnumber == iilo && best[2] != 0.0) {
    ct[1] = psum[1]/3;
    ct[2] = psum[2]/3;

    ctopilo=sqrt((ct[1]-p[iilo][1])*(ct[1]-p[iilo][1])+(ct[2]-p[iilo][2])*(ct[2]-p[iilo][2]));
    ctopinhi=sqrt((ct[1]-p[iinhi][1])*(ct[1]-p[iinhi][1])+(ct[2]-p[iinhi][2])*(ct[2]-p[iinhi][2]));
    ctopihi=sqrt((ct[1]-p[iihi][1])*(ct[1]-p[iihi][1])+(ct[2]-p[iihi][2])*(ct[2]-p[iihi][2]));

    if (ctopilo>ctopinhi && ctopilo>ctopihi) maxlength=ctopilo;
    else if (ctopinhi>ctopilo && ctopinhi>ctopihi) maxlength=ctopinhi;
    else maxlength=ctopihi;

```

```

if (maxlength<SIMPMAX && ctopilo/ctopinhi<1.7 && ctopilo/ctopihi<1.7) {
    /*Expand simplex if the newest becomes the lowest.*/
    for (i=1;i<=ndim;i++){
        xnew[i]=(5.0*p[ilo][i]-psum[i])/2.0;
    }
    phold1=p[ilo][1];
    phold2=p[ilo][2];
    fhold = fv[ilo];
    expand=1;
    p[ilo][1]=xnew[1];
    p[ilo][2]=xnew[2];
    rejnumber = ilo;
} else {
    /*Reflect the simplex away from the worst point.*/
    for (j=1;j<=ndim;j++) {
        xnew[j]=psum[j]-2.0*p[ihi][j];
    }
    p[ihi][1]=xnew[1];
    p[ihi][2]=xnew[2];
    rejnumber = ihi;
    expand=0;
}
} else {
    /*Reflect the simplex away from the worst point.*/
    for (j=1;j<=ndim;j++) {
        xnew[j]=psum[j]-2.0*p[ihi][j];
    }
    p[ihi][1]=xnew[1];
    p[ihi][2]=xnew[2];
    rejnumber = ihi;
}
}

/*Update the Valve settings based on simplex movement.*/
md->x[0] = xnew[1];
md->x[1] = xnew[2];

}
free_vector(psum,1,ndim);
free_vector(ct,1,ndim);
free_vector(xnew,1,ndim);
free_vector(ynew,1,ndim);
}
#endif GET_PSUM

```



## Appendix B: seqsimp.c

```
#define GET_PSUM for(j=1;j<=ndim;j++) {for(sum=0.0,i=1;i<=mpts;i++)sum+=p[i][j];psum[j]=sum;}

/***** performance function() *****/
* The performance function calculates the cost function from the target
* fibre fractions and the fibre fractions from the simulation. The
* equation is used to find the minimum.
*****/
float func(struct modeldat *md)
{
    float *y, *x, *k, cost;

    y = md->y;
    x = xglobal;
    k = kglobal;

    /*calculate the value of the performance equation.*/
    cost = 0.0;
    cost = 500*(RJRAT-X1SS)*(RJRAT-X1SS) + 500*(SHRAT-X2SS)*(SHRAT-X2SS);

    return(cost);
}

/***** simplex() *****/
* Sequential Simplex Method using symmetrical simplex and expansion
* The simplex function is a controller designed to find the optimum operating
* conditions of the dynamic model provided. The three points of the starting
* simplex are calculated and evaluated using the performance equation. A new
* simplex is determined based on the reflection of the old simplex away from
* the worst point.
* If the best point remains unchanged for 4 consecutive reflections, the size
* of the simplex is reduced.
*****/
void simplex(struct timing *t,struct modeldat *md)
/*Need the current rj and sh flow rates, and their corresponding fibre fractions.*/
{
    float value;
    float *y,*x,*ss,*k;
    struct simp *simp;
    int i,j,ilo,inh,ihi,mpts,ndim,simppt;
    float sum, *psum, *xnew, *ynew;
    float facrj, facsh;
    static int rejnumber;
    static float best[4],a,b;
    float rejected; /*Cost of the rejected point.*/
    float fin,med,cor;
    fin=1;
    med=1;
    cor=1;

    x = xglobal;
    ss = ssglobal;
    k = kglobal;
}
```

```

/*call the current values of the flow rates and fractions.*/
y = md->y;      /*the address of the states in the modeldat structure.*/
/*x = md->x;      /* " " " " inputs in the modeldat structure.*/

ndim = md->k[2]; /*the number of coordinates in each pt of the simplex.*/
mpts = ndim+1;
ilo = 1;
inhi = 1;
ihi = 1;

psum=vector(1,ndim);
xnew=vector(1,ndim);
ynew=vector(1,ndim);

/*calculate the changes to be made to the simplex.*/
a = SIMPSIZE/SIMPNP/sqrt(2)*(sqrt(SIMPNP+1)+SIMPNP-1);
b = SIMPSIZE/SIMPNP/sqrt(2)*(sqrt(SIMPNP+1)-1);

if (p[1][1] <=0 ) simppt=1;
else if (p[2][1] <= 0) simppt=2;
else if (p[3][1] <= 0) simppt=3;
else simppt=4;

if (simppt == 1){
    /*evaluate the performance equation for the first simplex point,
    and place the coordinates and perf in the simplex structure.*/
    value = func(md);

    /*indirect the pt number, inputs, performance value, best point.*/
    p[1][1] = RJRAT;
    p[1][2] = SHRAT;
    fv[1] = value;

    /* Determine the factors for the direction of the 2nd point of the
    starting simplex.*/
    if (p[1][1] > (LLRJRAT+ULRJRAT)/2) facrj = -1.0;
    else facrj = 1.0;

    if (p[1][2] > (LLSHRAT+ULSHRAT)/2) facsh = -1.0;
    else facsh = 1.0;

    /*calculate the coordinates of the second point of the simplex.*/
    md->x[0] = RJRAT + facrj*2*a;
    md->x[1] = SHRAT + facsh*b;

} else if (simppt == 2) {

    /*evaluate the performance equation for the second point.*/
    p[2][1] = RJRAT;
    p[2][2] = SHRAT;
    value = func(md);

    /*indirect the pt number, inputs, performance value, best point.*/
    fv[2] = value;

```

```

/* Determine the factors for the direction of the 3rd point of the
   starting simplex.*/
if (p[1][1] > (LLRJ RAT+ULRJ RAT)/2) facrj = -1.0;
else facrj = 1.0;

if (p[1][2] > (LLSH RAT+ULSH RAT)/2) facsh = -1.0;
else facsh = 1.0;

/*calculate the coordinates of the third point of the simplex.*/
md->x[0] = p[1][1] + facrj*2*b;
md->x[1] = p[1][2] + facsh*a;

} else if (simppt == 3) {

    /*evaluate the performance equation for the current point.*/
    p[3][1] = RJRAT;
    p[3][2] = SHRAT;
    value = func(md);

    /*indirect the pt number, inputs, performance value, best point.*/
    fv[3] = value;

    GET_PSUM

    /* Determine the ranking of each point in the simplex. The lowest (or
       best) will be labelled "ilo", the middle pt "inhi", and the high
       (or worst) pt "ihi". */
    ilo=1;
    ihi = fv[1]>fv[2] ? (inhi=2,1) : (inhi=1,2);
    for (i=1;i<=mpts;i++){
        if (fv[i] <= fv[ilo]) ilo=i;
        if (fv[i] > fv[ihi]) {
            inhi = ihi;
            ihi = i;
        } else if (fv[i] > fv[inhi] && i != ihi) inhi=i;
    }
    /*calculate the coordinates of the new point in the simplex.*/
    /*Reflect the simplex away from the worst point.*/
    for (j=1;j<=ndim;j++) {
        xnew[j]=psum[j]-2.0*p[ihi][j];
    }
    p[ihi][1]=xnew[1];
    p[ihi][2]=xnew[2];
    rejnumber = ihi;

    /*Update the Valve settings based on simplex movement.*/
    md->x[0] = xnew[1];
    md->x[1] = xnew[2];

} else {
    /*evaluate the performance equation for the current point.*/
    /*Find the cost of the current valve settings.*/
    value = func(md);
    rejected = fv[rejnumber]; /*Holder for the rejected cost.*/
    fv[rejnumber] = value; /*place the new cost into the rejected row.*/
}

```

## GET\_PSUM

```

/* Determine the ranking of each point in the simplex. The lowest (or
   best) will be labelled "ilo", the middle pt "inhi", and the high
   (or worst) pt "ihi". */
ilo=1;
ihi = fv[1]>fv[2] ? (inhi=2,1) : (inhi=1,2);
for (i=1;i<=mpts;i++){
    if (fv[i] <= fv[ilo]) ilo=i;
    if (fv[i] > fv[ihi]) {
        inhi = ihi;
        ihi = i;
    } else if (fv[i] > fv[inhi] && i != ihi) inhi=i;
}
/*Place the best pt into an extra matrix used to prevent the simplex
   from pivoting the best vertex for more than 4 iterations.*/
best[4] = best[3];
best[3] = best[2];
best[2] = best[1];
best[1] = fv[ilo];

/*calculate the coordinates of the new point in the simplex.*/
if (rejnumber==ihi) {
    /*newest pt is worse than most recent rejected, reject the 2nd worst pt.*/
    for (i=1;i<=ndim;i++) xnew[i]=psum[i]-2.0*p[inhi][i];
    p[inhi][1]=xnew[1];
    p[inhi][2]=xnew[2];
    rejnumber = inhi;

} else if (best[1]==best[2]&&best[2]==best[3]&&best[3]==best[4]&&best[1]!=0.0){
    /*simplex is cycling around one pt, shrink the simplex.*/
    if (SIMPSIZE <= SIMPMIN){
        /*don't shrink the simplex any further, let it continue to
           cycle around the optimum location.*/
        /*Reflect the simplex away from the best point.*/
        for (j=1;j<=ndim;j++) {
            xnew[j]=psum[j]-2.0*p[ilo][j];
        }
        p[ilo][1] = xnew[1];
        p[ilo][2] = xnew[2];
        rejnumber = ilo;
    } else {
        /*change the p matrix so the first row contains the middle pt.*/
        p[1][1]=p[inhi][1];
        p[1][2]=p[inhi][2];
        fv[1]=fv[inhi]; /*place the remaining cost into 1st spot.*/

        /*Reset the second and third rows of the P matrix*/
        for (i=1;i<=ndim;i++){
            p[2][i]=0.0;
            p[3][i]=0.0;
        }
        /*Reset the second and third places of the cost vector*/
        fv[2] = 0.0;
        fv[3] = 0.0;
    }
}

```

```

/*change the size of the simplex.*/
md->k[1] = 0.5*SIMPSize;

/*calculate the changes to be made to the simplex.*/
a = SIMPSize/SIMPnP/sqrt(2)*(sqrt(SIMPnP+1)+SIMPnP-1);
b = SIMPSize/SIMPnP/sqrt(2)*(sqrt(SIMPnP+1)-1);

/* Determine the factors for the direction of the 2nd point of the
starting simplex.*/
if (p[1][1] > (LLRJrAT+ULRJrAT)/2) facrj = -1.0;
else facrj = 1.0;

if (p[1][2] > (LLSHrAT+ULSHrAT)/2) facsh = -1.0;
else facsh = 1.0;

/*calculate the coordinates of the second point of the simplex.*/
xnew[1] = p[1][1] + facrj*2*a;
xnew[2] = p[1][2] + facsh*b;

best[1]=0.0;          /*Reset the best matrix.*/

rejnumber = 2;
}
} else if (rejnumber==ilo && best[2] != 0.0) {
/*Expand simplex if the newest becomes the lowest.*/
if (SIMPSize >= SIMPMax){
/*don't expand the simplex any further.*/
/*Reflect the simplex away from the worst point.*/
for (j=1;j<=ndim;j++){
xnew[j]=psum[j]-2.0*p[ihi][j];
}
p[ihi][1]=xnew[1];
p[ihi][2]=xnew[2];
rejnumber = ihi;
} else {
p[1][1]=p[ilo][1]; /*set best pt as first pt*/
p[1][2]=p[ilo][2]; /*in the new simplex */
fv[1] = fv[ilo];

for (i=1;i<=ndim;i++){
p[2][i]=0.0;
p[3][i]=0.0;          /*the simplex based on the best pt.*/
}
fv[2] = 0.0;
fv[3] = 0.0;

/*change the size of the simplex.*/
md->k[1] = 2.0*SIMPSize;

/*calculate the changes to be made to the simplex.*/
a = SIMPSize/SIMPnP/sqrt(2)*(sqrt(SIMPnP+1)+SIMPnP-1);
b = SIMPSize/SIMPnP/sqrt(2)*(sqrt(SIMPnP+1)-1);

/* Determine the factors for the direction of the
2nd point of the starting simplex.*/
if (p[1][1] > (LLRJrAT+ULRJrAT)/2) facrj = -1.0;

```

```

else facrj = 1.0;

if (p[1][2] > (LLRJ RAT+ULRJ RAT)/2) facsh = -1.0;
else facsh = 1.0;

/*calculate the coordinates of the second point of the simplex.*/
xnew[1] = p[1][1] + facrj*2*a;
xnew[2] = p[1][2] + facsh*b;

best[1]=0.0;          /*Reset the best matrix.*/

rejnumber = 2;
    }
} else {
    /*Reflect the simplex away from the worst point.*/
    for (j=1;j<=ndim;j++) {
        xnew[j]=psum[j]-2.0*p[ihi][j];
    }
    p[ihi][1]=xnew[1];
    p[ihi][2]=xnew[2];
    rejnumber = ihi;
}

/*Update the Valve settings based on simplex movement.*/
md->x[0] = xnew[1];
md->x[1] = xnew[2];

}
free_vector(psum,1,ndim);
free_vector(xnew,1,ndim);
free_vector(ynew,1,ndim);
}

#undef GET_PSUM

```

## Appendix C: simp.c

```
#define GET_PSUM for(j=1;j<=ndim;j++) {for(sum=0.0,i=1;i<=mpts;i++)sum+=p[i][j];psum[j]=sum;}

/***** performance function() *****/
* The performance function calculates the cost function from the target
* fibre fractions and the fibre fractions from the simulation. The
* equation is used to find the minimum.
*****/
float func(struct modeldat *md)
{
    float *y, *x, *k, cost;

    y = md->y;
    x = xglobal;
    k = kglobal;

    /*calculate the value of the performance equation.*/
    cost = 0.0;
    cost = -20*fabs(elong-eshort) + 2.0;
    cost += 0.055*(TARACFREE-acfree)*(TARACFREE-acfree);
    cost += 0.95+1.0*(1+tanh(6.0*(RJCONC-MAXRJCONS)))-3.5*(1+tanh(3.0*(RJCONC-2.65)));
    cost += 4.0-1.0*(1+tanh(100*(SHRAT-LLSHRAT)))-0.5*(1+tanh(100*(RJRAT-LLRJRAT)));
    cost += 1.5*(1+tanh(75*(SHRAT-ULSHRAT)))+1.0*(1+tanh(75*(RJRAT-ULRJRAT)));
    cost = 20*cost + 100;

    return(cost);
}

/***** simplex() *****/
* My method combining the ssm and n&m
* The simplex function is a controller designed to find the optimum operating
* conditions of the dynamic model provided. The three points of the starting
* simplex are calculated and evaluated using the performance equation. A new
* simplex is determined based on the reflection of the old simplex away from
* the worst point.
* If the best point remains unchanged for 4 consecutive reflections, the size
* of the simplex is reduced.
*****/
void simplex(struct timing *t,struct modeldat *md)
/*Need the current rj and sh flow rates, and their corresponding fibre fractions.*/
{
    float value;
    float *y,*x,*ss,*k;
    struct simp *simp;
    int i,j,mpts,ndim,simppt;
    float sum, *psum, *ct, *xnew, *ynew;
    float facrj, facsh, size, rjdif, shdif;
    static int rejnumber, expand, ilo, inhi, ihi;
    static float best[4],a,b,fhold, phold1, phold2;
    float rejected;
    float ctopilo, ctopinhi, ctopihi, minlength, maxlength;
    float fin,med,cor;
```

```

fin=1;
med=1;
cor=1;

x = xglobal;
ss = ssglobal;
k = kglobal;

/*call the current values of the flow rates and fractions.*/
y = md->y;      /*the address of the states in the modeldat structure.*/
/*x = md->x;      /* " " " " inputs in the modeldat structure.*/

ndim = md->k[2]; /*the number of coordinates in each pt of the simplex.*/
mpts = ndim+1;
ilo = 1;
inhi = 1;
ihi = 1;

psum=vector(1,ndim);
ct=vector(1,ndim);
xnew=vector(1,ndim);
ynew=vector(1,ndim);

/*calculate the changes to be made to the simplex.*/
a = SIMPSIZE/SIMPNP/sqrt(2)*(sqrt(SIMPNP+1)+SIMPNP-1);
b = SIMPSIZE/SIMPNP/sqrt(2)*(sqrt(SIMPNP+1)-1);

if (p[1][1] <= 0 ) simppt=1;
else if (p[2][1] <= 0) simppt=2;
else if (p[3][1] <= 0) simppt=3;
else simppt=4;

if (simppt == 1){
    /*evaluate the performance equation for the first simplex point,
    and place the coordinates and perf in the simplex structure.*/
    value = func(md);

    /*indirect the pt number, inputs, performance value, best point.*/
    p[1][1] = RJRAT;
    p[1][2] = SHRAT;
    fv[1] = value;

    /* Determine the factors for the direction of the 2nd point of the
    starting simplex.*/
    if (p[1][1] > (LLRJRAT+ULRJRAT)/2) facrj = -1.0;
    else facrj = 1.0;

    if (p[1][2] > (LLSHRAT+ULSHRAT)/2) facsh = -1.0;
    else facsh = 1.0;

    /*calculate the coordinates of the second point of the simplex.*/
    md->x[0] = RJRAT + facrj*2*a;
    md->x[1] = SHRAT + facsh*b;
} else if (simppt == 2) {
    /*evaluate the performance equation for the second point.*/

```



```

p[2][1] = RJRAT;
p[2][2] = SHRAT;
value = func(md);

/*indirect the pt number, inputs, performance value, best point.*/
fv[2] = value;

/* Determine the factors for the direction of the 3rd point of the
starting simplex.*/
if (p[1][1] > (LLRJ RAT+ULRJ RAT)/2) facrj = -1.0;
else facrj = 1.0;

if (p[1][2] > (LLSH RAT+ULSH RAT)/2) facsh = -1.0;
else facsh = 1.0;

/*calculate the coordinates of the third point of the simplex.*/
md->x[0] = p[1][1] + facrj*2*b;
md->x[1] = p[1][2] + facsh*a;

} else if (simppt == 3) {
/*evaluate the performance equation for the current point.*/
p[3][1] = RJRAT;
p[3][2] = SHRAT;
value = func(md);

/*indirect the pt number, inputs, performance value, best point.*/
fv[3] = value;

GET_PSUM

/* Determine the ranking of each point in the simplex. The lowest (or
best) will be labelled "ilo", the middle pt "inhi", and the high
(or worst) pt "ihi". */
ilo=1;
ihi = fv[1]>fv[2] ? (inhi=2,1) : (inhi=1,2);
for (i=1;i<=mpts;i++){
    if (fv[i] <= fv[ilo]) ilo=i;
    if (fv[i] > fv[inhi]) {
        inhi = ihi;
        ihi = i;
    } else if (fv[i] > fv[inhi] && i != ihi) inhi=i;
}
/*calculate the coordinates of the new point in the simplex.*/
/*Reflect the simplex away from the worst point.*/
for (j=1;j<=ndim;j++) {
    xnew[j]=psum[j]-2.0*p[ihi][j];
}
p[ihi][1]=xnew[1];
p[ihi][2]=xnew[2];
rejnumber = ihi;

/*Update the Valve settings based on simplex movement.*/
md->x[0] = xnew[1];
md->x[1] = xnew[2];

```

```

        expand=0;
    } else {
        /*evaluate the performance equation for the current point.*/
        /*Find the cost of the current valve settings.*/
        value = func(md);
        rejected = fv[rejnumber]; /*Holder for the rejected cost.*/
        fv[rejnumber] = value; /*place the new cost into the rejected row.*/

        /*if the expanded pt is worse than the previous pt.*/
        if(expand==1 && value>=rejected) {

            /*reverse the expansion*/
            p[rejnumber][1]=phold1;
            p[rejnumber][2]=phold2;
            fv[rejnumber]=fhold;
            /* Determine the ranking of each point in the simplex. The
            lowest (or best) will be labelled "ilo", the middle pt "inhi",
            and the high (or worst) pt "ihi". */
            ilo=1;
            ihi = fv[1]>fv[2] ? (inhi=2,1) : (inhi=1,2);
            for (i=1;i<=mpts;i++){
                if (fv[i] <= fv[ilo]) ilo=i;
                if (fv[i] > fv[ihi]) {
                    inhi = ihi;
                    ihi = i;
                } else if (fv[i] > fv[inhi] && i != ihi) inhi=i;
            }
            GET_PSUM
            /*and reject the worst pt from the unexpanded simplex.*/
            /*Reflect the simplex away from the worst point.*/
            for (j=1;j<=ndim;j++) {
                xnew[j]=psum[j]-2.0*p[ihi][j];
            }
            p[ihi][1]=xnew[1];
            p[ihi][2]=xnew[2];
            rejnumber = ihi;
            expand=0;
        } else {

            GET_PSUM

            /* Determine the ranking of each point in the simplex. The lowest (or
            best) will be labelled "ilo", the middle pt "inhi", and the high
            (or worst) pt "ihi". */
            ilo=1;
            ihi = fv[1]>fv[2] ? (inhi=2,1) : (inhi=1,2);
            for (i=1;i<=mpts;i++){
                if (fv[i] <= fv[ilo]) ilo=i;
                if (fv[i] > fv[ihi]) {
                    inhi = ihi;
                    ihi = i;
                } else if (fv[i] > fv[inhi] && i != ihi) inhi=i;
            }

            /*Place the best pt into an extra matrix used to prevent the simplex

```

```

    from pivoting the best vertex for more than 4 iterations.*/
    best[4] = best[3];
    best[3] = best[2];
    best[2] = best[1];
    best[1] = fv[i];

    /*calculate the coordinates of the new point in the simplex.*/
    if (rejnumber==ihi) {
        /*newest pt is worse than most recent rejected, reject the 2nd worst*/
        for (i=1;i<=ndim;i++) xnew[i]=psum[i]-2.0*p[ihi][i];
        p[ihi][1]=xnew[1];
        p[ihi][2]=xnew[2];
        rejnumber = ihi;

    } else if (best[1]==best[2]&&best[2]==best[3]&&best[3]==best[4]&&best[1]!=0.0){
        ct[1] = psum[1]/3;
        ct[2] = psum[2]/3;

        ctopilo=sqrt((ct[1]-p[i]lo)[1])*(ct[1]-p[i]lo)[1])+(ct[2]-p[i]lo)[2])*(ct[2]-p[i]lo)[2]));
        ctopinhi=sqrt((ct[1]-p[i]hi)[1])*(ct[1]-p[i]hi)[1])+(ct[2]-p[i]hi)[2])*(ct[2]-p[i]hi)[2]));
        ctopihi=sqrt((ct[1]-p[i]hi)[1])*(ct[1]-p[i]hi)[1])+(ct[2]-p[i]hi)[2])*(ct[2]-p[i]hi)[2]));

        if (ctopilo<ctopinhi && ctopilo<ctopihi) minlength=ctopilo;
        else if (ctopinhi<ctopilo && ctopinhi<ctopihi) minlength=ctopinhi;
        else minlength=ctopihi;

        if (minlength>SIMPMIN) {
            /*simplex is cycling around one pt, shrink the simplex.*/
            for (i=1;i<=ndim;i++){
                xnew[i]=0.25*(psum[i]+p[i]lo[i]);
            }
            p[i]lo[1]=xnew[1];
            p[i]lo[2]=xnew[2];

            best[1]=0.0;      /*reset the best matrix*/
            rejnumber = ilo;
        } else {
            /*Reflect the simplex away from the best point.*/
            for (j=1;j<=ndim;j++) {
                xnew[j]=psum[j]-2.0*p[i]lo[j];
            }
            p[i]lo[1]=xnew[1];
            p[i]lo[2]=xnew[2];
            rejnumber = ilo;
            expand=0;
            best[1] = 0.0;
        }
    }

    } else if (rejnumber == ilo && best[2] != 0.0) {
        ct[1] = psum[1]/3;
        ct[2] = psum[2]/3;

        ctopilo=sqrt((ct[1]-p[i]lo)[1])*(ct[1]-p[i]lo)[1])+(ct[2]-p[i]lo)[2])*(ct[2]-p[i]lo)[2]));
        ctopinhi=sqrt((ct[1]-p[i]hi)[1])*(ct[1]-p[i]hi)[1])+(ct[2]-p[i]hi)[2])*(ct[2]-p[i]hi)[2]));
        ctopihi=sqrt((ct[1]-p[i]hi)[1])*(ct[1]-p[i]hi)[1])+(ct[2]-p[i]hi)[2])*(ct[2]-p[i]hi)[2]));

        if (ctopilo>ctopinhi && ctopilo>ctopihi) maxlength=ctopilo;
    }

```

```

else if (ctopinhi>ctopilo && ctopinhi>ctopihi) maxlength=ctopinhi;
else maxlength=ctopihi;

if (maxlength<SIMPMAX) {
    /*Expand simplex if the newest becomes the lowest.*/
    for (i=1;i<=ndim;i++){
        xnew[i]=(5.0*p[ilo][i]-psum[i])/2.0;
    }
    phold1=p[ilo][1];
    phold2=p[ilo][2];
    fhold = fv[ilo];
    expand=1;
    p[ilo][1]=xnew[1];
    p[ilo][2]=xnew[2];
    rejnumber = ilo;
} else {
    /*Reflect the simplex away from the worst point.*/
    for (j=1;j<=ndim;j++) {
        xnew[j]=psum[j]-2.0*p[ihi][j];
    }
    p[ihi][1]=xnew[1];
    p[ihi][2]=xnew[2];
    rejnumber = ihi;
    expand=0;
}
} else {
    /*Reflect the simplex away from the worst point.*/
    for (j=1;j<=ndim;j++) {
        xnew[j]=psum[j]-2.0*p[ihi][j];
    }
    p[ihi][1]=xnew[1];
    p[ihi][2]=xnew[2];
    rejnumber = ihi;
}
}

/*Update the Valve settings based on simplex movement.*/
md->x[0] = xnew[1];
md->x[1] = xnew[2];

}
free_vector(psum,1,ndim);
free_vector(ct,1,ndim);
free_vector(xnew,1,ndim);
free_vector(ynew,1,ndim);
}

#undef GET_PSUM

```